NBSIR 75-692

A Minicomputer-Based System for the Measurement and Analysis of Community Noise

R. L. Fisher, D. S. Blomquist, J. S. Forrer, D. M. Corley

Mechanics Division
Institute for Basic Standards
National Bureau of Standards
Washington, D. C. 20234

May 1975

Final Report

Prepared for

Bioacoustics Division
U. S. Army Environmental Hygiene Agency
Aberdeen Proving Ground, Maryland 21010

NBSIR 75–692

A MINICOMPUTER-BASED SYSTEM FOR THE MEASUREMENT AND ANALYSIS OF COMMUNITY NOISE

R. L. Fisher, D. S. Blomquist, J. S. Forrer, D. M. Corley

Mechanics Division
Institute for Basic Standards
National Bureau of Standards
Washington, D. C. 20234

May 1975

Final Report

Prepared for
Bioacoustics Division
U. S. Army Environmental Hygiene Agency
Aberdeen Proving Ground, Maryland 21010

U.S. DEPARTMENT OF COMMERCE, Rogers C.B. Morton, Secretary

NATIONAL BUREAU OF STANDARDS, Richard W. Roberts, Director

TABLE OF CONTENTS

A Minicomputer-Based System for the Measurement and Analysis of Community Noise

R. L. Fisher
D. S. Blomquist
J. S. Forrer
D. M. Corley

An operating system for the measurement and analysis of community noise was turned over to the Army for their use in February 1975, thus accomplishing the transfer of technology developed by NBS to the Bioacoustics Division, U. S. Army Environmental Hygiene Agency. This report documents the hardware and software packages prepared by NBS in support of this system.

1. INTRODUCTION

The Applied Acoustics Section, Mechanics Division, of the National Bureau of Standards developed a specialized minicomputer-based system for the measurement and analysis of community noise. Since the system originally was developed for in-house use, the hardware and software were in a form that could only be fully utilized by members of the section who were intimately familiar with the operation of the system.

The Bioacoustics Division (BAD) of the U. S. Army Environmental Hygiene Agency, Aberdeen Proving Grounds, required similar measurement and analysis capability to effectively carry out its mission; however, their system would have to be in a form that could be used by personnel not possessing an in-depth familiarity with measurement systems or computer software.

In May 1974, a program was initiated at NBS under BAD sponsorship. The overall objective of this program was the transfer of the technology of measurement and analysis of community noise developed by NBS to BAD. NBS provided the technical expertise in assisting BAD in assembling and documenting such a system and provided the following hardware and software packages: (1) software for sampling community noise levels, (2) software for processing community noise data, (3) utility software programs, (4) community noise interface hardware, and (5) real time analyzer interface hardware and appropriate software driver.

An operating system was turned over to the Army for their use in February 1975. This report documents the hardware and software packages developed by NBS. The documentation includes:

- explanation of the hardware and its interaction with the software,
- block diagrams to aid understanding of the system and facilitate any future system modifications,
- sample calculations for testing and diagnostics,
- user instructions.

2. SYSTEM DESCRIPTION

The community noise measurement system developed by NBS for the Bioacoustics Division was designed to be essentially a "turn-key" data acquisition and analysis system. The hardware portion of the system is comprised of commercially produced analog equipment (microphones, measuring amplifiers, etc.), an interface designed and fabricated by NBS, and an off-the-shelf minicomputer together with its digital peripheral devices.

The input to the present system consists of two microphones which monitor the acoustic environment to be studied and an optional wind sensor for monitoring wind speed during data acquisition.[1/]

Once the proper programs (see Section 4.1. for description of software and loading instructions) are loaded, the minicomputer operates as a hardware controller, sampling and storing the A-weighted sound levels (in decibels re 20 μPa) for both acoustic data channels every 0.1 second. After a minute of data (600 samples) has been stored, the minicomputer writes the data on IBM[2/] compatible digital magnetic tape for later analysis. It should be noted that a 731.5 metre (2400 foot) digital magnetic tape can store approximately 50 hours of community noise data. Every ten minutes the minicomputer prints out a summary line of statistical descriptors of the community noise -- L_1, L_{10}, L_{50} and L_{eq}[3/] (for a ten minute interval) -- on the Teletype, thus providing an indication of overall system operation. These line summaries which include the tape number, time of day, A-weighted sound levels, etc., are formatted such that they can conveniently serve as the basis for an accurate data log.

The complete data tapes can be processed on a Univac 1108 computer system which utilizes FORTRAN programs (described in Section 4.2.). Once analysis is complete, the community noise data can be produced in tabular and/or graphical form.

1/ The system design provides the capability for expansion to a maximum of eight channels of analog data.

2/ Certain commercial equipment, instruments, or material are identified in this report in order to adequately specify the BAD system. In no case does such identification imply recommendation or endorsement by the National Bureau of Standards, nor does it imply that the material or equipment identified is necessarily the best available for the purpose.

3/ L_1 is the A-weighted sound level which is exceeded 1 percent of the time. L_{10} is the A-weighted sound level which is exceeded 10 percent of the time. Both L_1 and L_{10} are often used to represent the higher-level, shorter-duration sounds. L_{50} is the A-weighted sound level which is exceeded 50 percent of the time. L_{eq} is the energy equivalent A-weighted sound level.

Except for brief periods (generally 7 seconds every 10 minutes) while the Teletype is printing out the data line summaries, the A-weighted sound level (in decibels re 20 μPa multiplied by a factor of ten) for either channel 1 or channel 2 (selectable by a switch on the front panel of the interface) is digitally displayed (four digit display) on the front panel of the interface for the convenience of the experimenter. These A-weighted sound levels will typically agree very closely with the analog meter readings of the measuring amplifier for the appropriate channel and therefore provides on the spot confirmation that the procedures for system calibration were correctly followed.

The computer interface designed and fabricated by NBS also contains a time-of-day clock, which the minicomputer interrogates once each minute (and stores the time to the nearest second), and a Brüel and Kjaer one-third-octave-band real-time-analyzer controller which is functionally independent from the community noise system.

Using a software driver (see Section 4.1.3) the real time analyzer can be sampled to obtain the root mean square (rms) value of the level in each one-third octave band from 12.5 Hz to 40 kHz (if these filters exist in the real time analyzer), a selected weighted sound level and the overall sound pressure level (linear) at the output of the analyzer in binary coded decimal (BCD) code. A FORTRAN controlling program (not supplied by NBS) could be written to store these data and dump them on to digital magnetic tape for later processing and plotting by a large computer system.

The four digit hexidecimal display on the front panel of the interface which was discussed earlier is independent (hardware wise) from the remainder of the system; therefore, it can be utilized to display up to four hexidecimal digits of information on command from the minicomputer.

Field experience pointed out the necessity for a fast-acting 117 v A.C. power line conditioner to protect the system from power transients and offset the effect of power brownouts on the computer system. This feature was included in the BAD system. In the event of a longer term (greater than a few milliseconds) power outage that cannot be handled by the conditioner, a power-fail-safe software driver was written by NBS to provide, in conjunction with minicomputer hardware, for orderly shutdown of the central processing unit (CPU) during power failure and a resumption of the program when power is restored. Long term power outages result in the complete shutdown of the minicomputer's digital magnetic tape system. The power-fail-safe hardware-software cannot restart the tape system after it shuts down. When the tape unit goes down one must follow the normal manual start up procedures.

The remainder of this report will concentrate on the details of the hardware and software packages developed by NBS.

3. DESCRIPTION OF INTERFACE HARDWARE

The system developed for the Bioacoustics Division has two channels for acoustic data and one channel for wind data. The two acoustic channels have an input impedance of 68 KΩ and have a voltage range of 0 to 4.5 volts. The windsensor input has an input impedance of 500 KΩ and an voltage range of 0 to 4 volts. The windsensor input has RC integration for smoothing of data. The time constant is 500 milliseconds. The windsensor is a small full wave rectified alternating current generator driven by a conventional 3 bladed anemometer.

Additional channels may be added by inclusion of a buffer amplifier, sample and hold module, timing changes and software changes for each channel added. Maximum expansion of the system -- eight channels -- would also require an expansion of the interface memory capacity.

In addition to the community noise measurement system, the interface also contains the necessary logic to allow a Brüel and Kjaer one-third octave band real time analyzer (model 3347) to be coupled to the minicomputer.

This section contains a simplified explanation of interface operation. For clarity, the minicomputer as well as the commercial analog equipment are shown in the block diagram (Figure 1) describing interface components.

The input signal from the RMS detector log converter is the logarithm of the A-weighted root mean square (rms) value of the acoustic signal and is buffered by the input amplifier. The buffered signal is fed to a sample and hold module. System timing and control is maintained by a programmable read only memory (PROM). This PROM is addressed by an eight stage counter which is controlled by the crystal controlled oscillator in the time-of-day clock. The PROM strobes the sample and hold module ten times per second. The output of the sample and hold module is fed to a multiplexer (MUX) and then through a buffer amplifier to an analog to digital converter (ADC).

The timing sequence is as follows:

1. The sample and hold module is strobed and the data are held on the storage capacitor.

2. After the data have settled on the hold capacitor (10 RC time constants) the input channels are multiplexed to the ADC.

3. The ADC is strobed to start conversion.

The analog signal is converted to a 10 bit digital word by the ADC. The 10 bit ADC results in a quantization error of 0.05 dB when the analog signal is sent to a Brüel and Kjaer measuring amplifier type 2607 and the log, dc, output is connected to the interface. The end of conversion pulse from the ADC stores the digitized voltage and a channel number in a 12 bit by 320 word asynchronous first-in/first-out (FIFO) memory. The purpose of the memory is to allow the computer enough time to perform the necessary calculations and "house keeping" without loss of data. The minicomputer is interfaced to the FIFO memory on an interrupt basis.

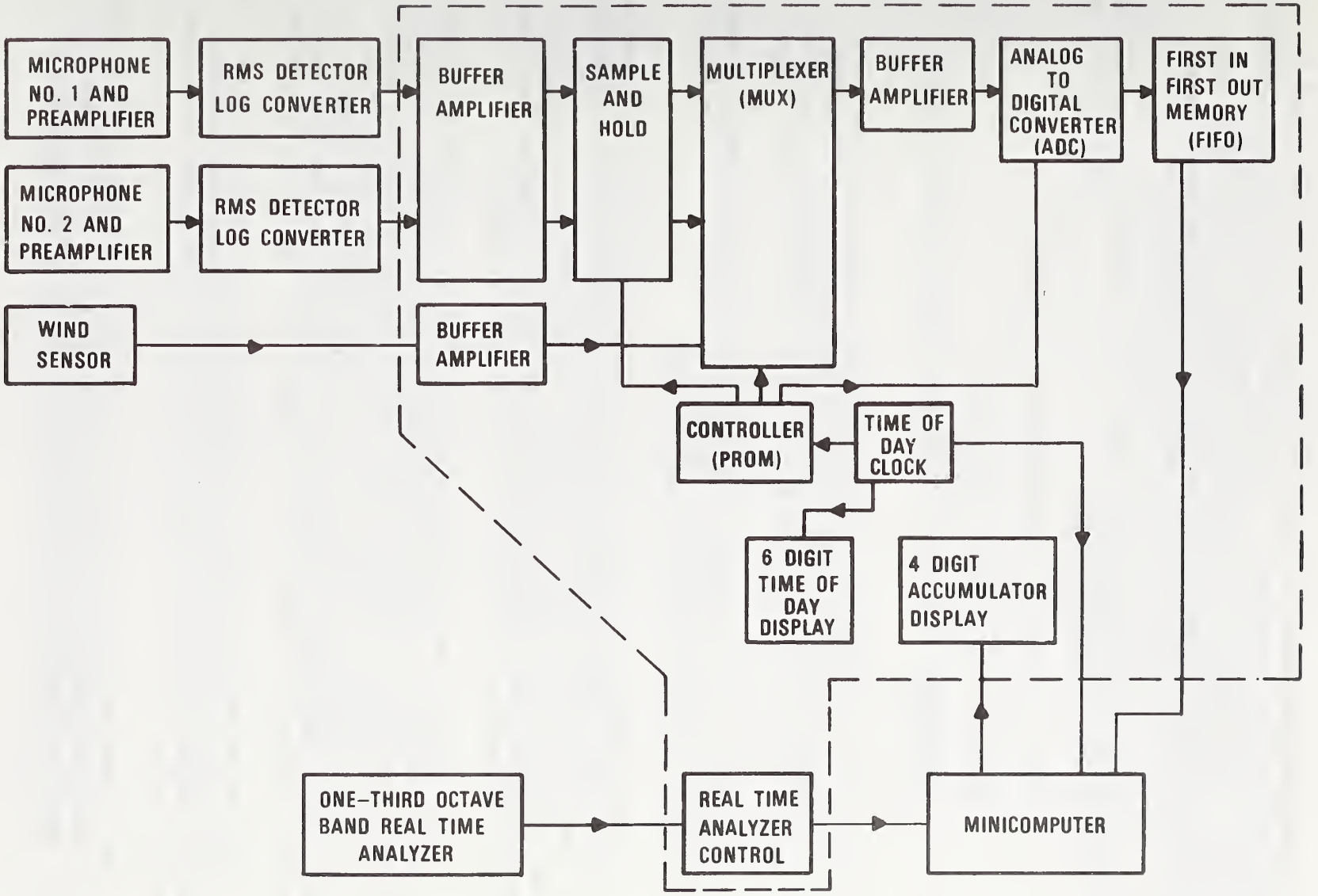


Figure 1. Block diagram showing community noise measurement system with emphasis on the functions of the interface (shown within the dashed lines) designed and fabricated by NBS.

The system also includes a CMOS time-of-day clock, which is read by the minicomputer once each minute. The time base for the clock is crystal controlled. The format of the clock is 24 hours with six BCD digits for hours, minutes and seconds. The clock has a battery powered standby mode which allows for clock operation for a minimum of 1 hour without external power.

The accuracy of the time-of-day clock is governed by the tolerances supplied by the crystal manufacturer:

1. Initial value within 5 ppm (parts per million).

2. Aging per year equals 2 ppm or less.

3. Temperatures from 10°C to 40°C will cause additional errors up to 8 ppm.

Using a worst case assumption for each of the three factors (15 ppm), the total error in one year if the clock were not reset would be 7.9 minutes (about 1.3 seconds per day). After the crystal is one year old, the total drift over a 48 hour period at 20°C would be less than 0.6 seconds (7 ppm).

In December 1974 the crystal oscillator in the clock in the BAD interface was found, in actual laboratory tests, to be accurate to within 5 ppm of the nominal specified frequency value at 20°C.

As stated earlier, the system also has the necessary logic for reading out the data from a Brüel and Kjaer one-third octave band real time analyzer (RTA). The 3 1/2 digits from the analyzer are accompanied by two status bits for indicating underrange and overload. In this case, underrange is defined as a level that is less than the RTA's internal reference level while overload is defined as an overload of the analog input amplifier.

On the front of the interface panel there are two displays which allow for operator control -- a time-of-day display and an accumulator display. The time-of-day display shows the time of day in hours, minutes and seconds on a six digit display in 24 hour format. The accumulator display shows the accumulator contents (16 bits) on a four digit hexidecimal readout. A single computer instruction is required for the display and the data are set in latches such that the data will be displayed until new data are received.

The time of day is set by use of the hold, slow and fast switches. The hold switch stops the clock, the slow switch advances the clock at a one minute per second rate and the fast switch advances the clock at a one hour per second rate.

Front panel switches also include an external sense switch for computer control and a power switch to disable the battery power for the time-of-day clock. When the computer is to be turned off for more than one hour, the

battery power should also be turned off to avoid completely discharging the batteries.

There are three switches on the back of the interface. The purpose of these switches is for trouble shooting the hardware and software. All the switches are debounced to eliminate multiple pulses. The Master Reset switch clears the interface memory of all data and resets the PROM to the first location. There are two switches for either shifting data into or out of the memory.

4. DESCRIPTION OF SYSTEM SOFTWARE

This section and the appendices document the FORTRAN and SYM II (assembly language for this particular 16-bit minicomputer [Raytheon 704]) assembly language software for the minicomputer-based community noise system and real-time analyzer interface system as well as the FORTRAN software for the analysis of community noise tapes. The documentation includes program listings, overall block diagrams, test programs to be utilized for diagnosis of system problems and typical examples of the output from the main FORTRAN programs.

4.1. Fortran and Assembly Language Software for the Minicomputer-Based Community Noise System and Real-Time Analyzer Interface System.

The operation of the minicomputer-based data acquisition system is discussed from a software point of view in this section. The level of explanations assumes familiarity with programming in conversational FORTRAN and SYM II[4/] assembly language.

In general, two main FORTRAN programs and their respective subroutines must be loaded into the minicomputer and be operable in order to be able to acquire a possible eight channels of data on digital magnetic tape (for later analysis utilizing a large computer system such as the Univac 1108 at the National Bureau of Standards). It should be noted that substantial changes to both the hardware interface and to the software would be necessary for expanding the present system (2 channels) to a maximum of eight channels.

The first main program is called "CHECK". Once the "CHECK" program is loaded (see Section 4.1.5. for loading procedures) the measuring amplifier (Brüel and Kjaer 2607) provides the output for acoustic data channel 1 and is manually stepped over its entire 50 dB range and a linear fit is performed on the resulting digital values received by the minicomputer from the community noise interface using the method of least squares. This procedure determines the factor relating analog A-weighted sound levels in decibels (re 20 μPa) to the digital values transferred to the minicomputer via the community noise interface. The process is repeated for acoustic data channel 2 and a calibration paper tape is automatically punched out for use in the "NOISE" program.

The second main program, called "NOISE", (1) receives and buffers the digital values which are proportional to the output values of the measuring amplifier, (2) converts these values to exact decibel values and (3) writes the community noise data on digital tape. The reference decibel value is maintained in the program by periodic application of a pistonphone calibration on the microphone for each acoustic data channel.

If either of the two main FORTRAN programs malfunction, the Raytheon supplied diagnostic programs[5/] -- CPU 30, 31, etc. -- and the system interface

[4/] "Raytheon 704 Users Manual"
Raytheon Data Systems
1415 Boston-Providence Turnpike
Norwood, Massachusetts 02062
page 8.12

[5/] Raytheon 704 Users Manual, page 8.84

test programs written by NBS (outlined in Section 4.1.3.) should be run to determine whether the fault lies in the central processing unit (CPU) or the interface.

4.1.1. Conversational FORTRAN Programs for Calibration and Use of the Community Noise Measurement System

The measuring amplifier (Brüel and Kjaer 2607) used in the BAD community noise system has a dc output that is directly proportional to the logarithm of the root mean square (rms) input voltage. This dc output is connected to an analog multiplexer (for switching channels) and then to an analog to digital converter in the community noise interface. Therefore, the digital output of the interface is also directly proportional to the logarithm of the rms input voltage of the measuring amplifier. Once the values for f and c in the following equation have been determined, the "NOISE" program will directly compute the A-weighted sound level (SPL) in decibels (re 20 μPa) from the raw digital values transferred from the interface to the minicomputer.

$$SPL = fD + c \qquad (1)$$

where D = raw digital value from the community noise interface
f = slope of the least squares curve fit discussed in Section 4.1.
c = constant based on the pistonphone level.

The conversational FORTRAN program called "CHECK" in conjunction with the manual switching of the input attenuator of the measuring amplifier (previously discussed in Section 4.1.) establishes the f values for both acoustic data channels. (See appendices A and B for program listings). The procedure to use is as follows:

1. Load "CHECK" program using one of the methods outlined in Section 4.1.5. of this report.

2. Follow the directions provided by program "CHECK" which are typed on the Teletype.

3. Save the listing produced by the "CHECK" program for future reference. Save the paper tape punched for later use by the "NOISE" program.

The conversational FORTRAN subroutine "ORTH" is a least squares curve fitting routine using orthogonal polynomials. The "CHECK" program calls subroutine "ORTH" to perform a first order (linear) fit on the array of measuring amplifier input values (0 to 50 dB) versus an array of raw interface value (0 to 1023) solving for c and f in equation (1). The values of f for both acoustic data channel 1 and acoustic data channel 2 are punched on a paper tape (along with the year, day and time) by the "CHECK" program for later use in the "NOISE" program. The values for c are not stored because c will be determined by pistonphone calibration and measuring amplifier gain settings in the "NOISE" program. The flow diagram for the "CHECK" program is presented in Appendix A.

The "NOISE" program conducts a two-channel community noise survey by sampling the acoustic data channels and the optional wind data channel at a 10 Hz rate. Once a minute, the noise data (A-weighted sound level in decibels re 20 μPa) and the maximum wind speed are written on magnetic tape for later analysis.

The procedure for using the "NOISE" program is as follows:

1) Load "NOISE" program into the minicomputer using one of the methods shown in Section 4.1.5. of this report.

2) Follow the directions given by the "NOISE" program in the Teletype printout.

3) After the program is routinely recording data on magnetic tape, a new pistonphone calibration can be initiated by putting sense switch 1 up on the CPU. Similarly putting sense switch 2 up will request a new 2607 gain setting.

4) Save the listing produced by the "NOISE" program as a data log for the data stored on the magnetic tape.

The "NOISE" program will print out L_1, L_{10}, L_{50} and L_{eq} every ten minutes on the system Teletype. (These values are not recorded on the magnetic tape). The Teletype printout forms a valuable record of the data recorded on the magnetic tape. L_1 is a useful guide for determining the proper input attenuator settings for the measuring amplifier. Set the measuring amplifier attenuators so that the value of L_1 is within 10 dB of full scale on the measuring amplifier. These L_i values (calculated on fixed 10 minute intervals) will not agree exactly with L_i values calculated by a box car averaging method in the program described in Section 4.2. The flow diagram for the "NOISE" program is presented in Appendix A.

"TOGLE" is a conversational FORTRAN subroutine used by both the "CHECK" program and the "NOISE" program to detect sense switch 0 changes signaling a user response.

4.1.2. Assembly Language Subroutines

a. Subroutine "STEM 1"

Main programs such as "CHECK" and "NOISE" and their subroutines can be added to the normal Raytheon systems tape as described in Section 4 of the Raytheon Magnetic Tape Operating System Manual[6/]. Developing this normal systems tape is time consuming and calling the required programs under "XRAY" control would be a lengthy process. To avoid this problem and in addition avoid operating exclusively with long paper tapes the "STEM 1" program was written. (See appendix B for program testing.)

6/ "Magnetic Tape Operating System"
Raytheon Data Systems
1415 Boston-Providence Turnpike
Norwood, Massachusetts 02062

When called by a conversational FORTRAN program, "STEM 1" checks to see if sense switches 0, 1, and 2 are up. If these switches are up and the magnetic tape is "write enabled", a small loader and a complete core image dump will be written on magnetic tape thus developing a "special" systems tape. The program on the special systems tape can be called into memory by:

1. Mounting special tape in the magnetic tape drive and bringing the tape to load point (BOT) and on line.

2. Push RESET on the CPU.

3. Push magnetic tape LOAD button N times where N equals 1 plus 2 times the number of programs ahead of the desired program on magnetic tape.

4. Push RUN on CPU.

5. Push RUN on CPU.

b. Subroutine "DATA"

This program establishes interrupt level 5 linkage which allows data to be brought into the computer from the interface. (Subroutine "INTL" must also be called to enable interrupt level 5 thereby starting actual data transfer.) "DATA" makes the instantaneous values of the wind channel, acoustic data channel 1 and acoustic data channel 2 available for use in the main FORTRAN program at a 10 Hz rate. Program "CHECK" utilizes the "DATA" subroutine to input raw interface data for curve fitting and to send these data to the interface display. Program "NOISE" utilizes the "DATA" subroutine only for pistonphone calibration and does not send any data to the interface display. (Program "NOISE" uses subroutine "DISP" to send A-weighted sound level data to the display.)

c. Subroutine "LFD"

Program "NOISE" cannot convert raw interface values to actual A-weighted sound levels in decibels (re 20 μPa) while the Teletype is typing. Therefore, typing time must be minimized. By calling subroutine "LFD", the "NOISE" program can space up to ten lines per second instead of two lines per second when utilizing normal Raytheon formatting. When called by a conversational FORTRAN program, subroutine "LFD" sends a line feed character to the Teletype causing the paper to space up by one line.

d. Subroutine "POWF5"

This subroutine provides for the orderly shutdown of the central processing unit (CPU) with no program loss in the event of power interruption. When power is restored, "POWF5" sends a master reset signal to the interface and the system returns to the program that was in progress when the power failure occurred. If the power is off for more than a few milliseconds, the magnetic

tape unit will shut down and manual intervention is required to prepare it for operation. If bit 5 of word X'4B' was set prior to the power failure (by "INTL" subroutine), "POWF5" would reenable interrupt level 5 when power was restored.

e. Subroutine "MULTI"

The "NOISE" program utilizes subroutine "MULTI" to allow the values for the wind channel and the two acoustic data channels to be transferred from the interface to the minicomputer during actual data collection. (Program "NOISE" utilizes subroutine "DATA" to bring in raw data during pistonphone calibration.) Subroutine "MULTI" fills a 1200 word array -- one minute of data -- with raw data from acoustic data channels 1 and 2. Subroutine "INTL" must be utilized to enable interrupt level 5 before subroutine "MULTI" can transfer data from the interface to the minicomputer.

f. Subroutine "MMON"

Subroutine "MMON" modifies the Raytheon supplied operating software so that it will not continuously write out MO, MO, ... on the Teletype when the magnetic tape unit shuts down.

g. Subroutine "DMAMAG"

Subroutine "DMAMAG" checks the direct memory access (DMA) magnetic tape unit status and returns the status to the calling conversational FORTRAN program. It is particularly useful for locating end of tape (EOT) and for determining whether or not the magnetic tape is "write enabled".

h. Subroutine "INTL"

Subroutine "INTL" is used to enable or disable any one of the interrupt levels and to record the enable status of each level in the enable status word in location X'4B'. Subroutine "POWF5" uses this status word following a power failure to determine if interrupt level 5 should be reenabled.

i. Subroutine "DISP"

This subroutine has two modes of operation. In the first mode, "DISP" converts a binary number to binary coded decimal (BCD) format and sends it to the interface display. In the second mode, "DISP" converts the number to BCD and sends it to the interface display only if the external sense switch setting agrees with a calling parameter JCHAN (see program listing in Appendix B). The "NOISE" program calls subroutine "DISP" to display A-weighted sound level data in decibels (re 20 μPa). The person performing the data analysis can elect to display either acoustic data channel 1 or acoustic data channel 2 by placing the external sense switch up or down respectively.

j. Subroutine "LEADR"

When called by a main conversational FORTRAN program, this subroutine will punch a leader on the Teletype. The "CHECK" program calls subroutine "LEADR" to punch a leader in front of and behind the calibration tape it produces for use in the "NOISE" program.

k. Subroutine "CLOK"

When called by a main program, subroutine "CLOK" reads the interface clock until the time agrees for two successive interrogations. The time in hours, minutes and seconds is then utilized by the main program as data.

Program listings for subroutines discussed in this Section are contained in Appendix B.

4.1.3. Test Programs for the Community Noise/Real Time Analyzer Interface (Plus Real Time Analyzer Subroutine Driver)

a. Absolute Binary Routine "MINOS1"[7/]

This program is used for system checkout and for performing elementary tests on:

1) interface display

- external sense switch up for a decimal display.

- external sense switch down for a hexidecimal display.

2) wind channel

- sense switches 1 and 2 down to display raw wind values (not corrected to knots).

3) acoustic data channel 1

- sense switch 1 (SS1) up (SS2 down) to display raw channel 1 values (not converted to decibels).

4) acoustic data channel 2

- sense switch 2 (SS2) up (SS1 down) to display raw channel 2 acoustic values (not converted to decibels).

5) master reset

- sense switch 3 up to provide a master reset command to the interface after each number is displayed.

6) interrupt level 5.

Note: Put sense switch 0 up to halt the CPU after each value is displayed if desired. (Push RUN to continue.)

7/ The program listing for MINOS1 and the remaining programs in Section 4.1.3. are in Appendix C.

The "MINOS1" program is loaded into the minicomputer by giving the XRAY directive: AL. If the word FACE remains on the interface display more than a fraction of a second after loading the program, something is wrong with interrupt level 5. Either the interface is not sending an interrupt pulse to the CPU or the interrupt card in the CPU is bad.

b. Conversational FORTRAN Program "TCLOK"

"TCLOK" is a test program utilized for printing out the current time read from the interface clock. Program "TCLOK" calls subroutine "CLOK" to transfer the hours, minutes and seconds values from the display to the Teletype.

An example of the printout from "TCLOK" is shown in Section 4.1.4.

c. Conversational FORTRAN Program "TRTA1"

The "TRTA1" program receives values for the sound pressure level in each one-third octave band, a selected weighted sound level and the value for the overall sound pressure level (linear) in BCD format from the output of the real time analyzer (RTA) by calling the RTA driver subroutine "RTA1". The "TRTA1" program prints out these values along with a channel underflow bit on the Teletype. If the underflow bit is a 1, that particular channel is at or below the base line value of the real time analyzer. If the underflow bit is a 0, the value is above the baseline. If the input amplifier of the real time analyzer was overloaded during the time period that data were sampled and sent to the interface, an overload message will be printed out.

An example of "TRTA1" output is shown in Section 4.1.4.

d. Subroutine "RTA1"

When called by a conversational FORTRAN program, subroutine "RTA1" commands the real time analyzer to digitize the values in each one-third octave band, the selected weighted band and the overall (linear) band and to send the values in BCD format to the "RTA1" program as data. Subroutine "RTA1" converts these BCD numbers to binary numbers (decibel times a factor of ten) and sends them to the main conversational FORTRAN program as fixed point integers. Channel underflow bits and the input attenuator overload values are also sent to the FORTRAN main program.

4.1.4. Typical Output Listings

This section contains output listings from the "CHECK", "NOISE", "TCLOK", and "TRTA1" programs.

The portion of the output listing for the "CHECK" program shown below should be kept in a log book since these calibration data establish the system accuracy. The year, date and time of the calibration is printed in the upper right hand corner of the printout and also appears at the beginning of the punched paper tape output which will be read by the "NOISE" program. The first three columns in the middle of the listing show the five step 50 dB dynamic range of the Brüel and Kjaer measuring amplifier (type 2607), values for acoustic data channel 1 (raw interface numbers) and values for acoustic data channel 2 (raw interface numbers). These data are utilized as the basis for the linear curve fitting routine discussed in Sections 4.1. and 4.1.1. The fourth and fifth columns show the difference in decibels between the 2607 values and the values determined by the curve fitting routine. In the example, the maximum system error was 0.12 dB. The linear equations for acoustic data channels 1 and 2 are printed out and the factor relating raw interface values (ADC values) to 2607 values are also punched out on the special output paper tape that is read by the "NOISE" program.

```
2607 - INTERFACE NOISE SYSTEM LINEARITY CHECK

                                                    YEAR   = 1975
                                                    DAY    =   58
                                                    HOUR   =    2
                                                    MINUTE =   23

    2607      CH1       CH2       CH1       CH2
     DB       ADC       ADC       DB        DB
            VALUE     VALUE     DIFF      DIFF

      0         5         5      .11      -.00    (0 DB VALUE NOT USED)
     10       204       205      .07       .11
     20       400       398     -.12      -.12
     30       603       597      .03      -.05
     40       803       796      .03       .01
     50      1002       994     -.01       .03


    CH1 DB = CONSTANT + 50.02387E -3   * (ADC VALUE)
    CH2 DB = CONSTANT + 50.60568E -3   * (ADC VALUE)


TURN ON PUNCH, TOGGLE SENSE SWITCH 0
   1975  58  2 23  50.02387E -3  50.60568E -3
TURN OFF PUNCH, TOGGLE SENSE SWITCH 0 TO RESTART
CHECK OUT IS COMPLETED. USE PAPER TAPE IN NOISE
PROGRAM IF ABOVE RESULTS ARE SATISFACTORY.
```

An example printout of the "NOISE" program is presented on the next page. The second line of the heading contains the page number, magnetic tape number, year, and date and time of the calibration. The time of calibration refers to the time the calibration paper tape was prepared. Below the heading, there are sixteen labeled columns of numbers that make up the main body of the listing and serve as the basis for an accurate data log of the noise levels measured as well as certain checks on the minicomputer system. The first four columns give the day of the year and the time (hours, minutes, seconds) corresponding to the end of the 10 minute block of data. The next four columns show the L_1, L_{10}, L_{50} and L_{eq} values for acoustic data channel 1.[8/] The next four columns show similar data for acoustic data channel 2. The column labeled WIND shows the highest wind speed (in knots) for the entire 10 minute block. The column labeled AC PFS is a count of the number of ac power failures that occurred during the 10 minute block. The column labeled # ERR is a count of the number of incorrect channel numbers generated by the interface during the 10 minute block. Finally, the column labeled # SKIP is a count of the number of missing or skipped channel numbers during the 10 minute block. Non-zero values for either # ERR or # SKIP indicate interface malfunction unless AC PFS is also non-zero in the same time block. In the example shown, an attenuator change was made on the measuring amplifier between data blocks 3 and 4. The new attenuator settings for channels 1 and 2 were .01 volts. A new pistonphone calibration was also made after data block 4. The system drift, in this case, was 0.1 dB.

An example of the printout of the "TCLOK" program is reproduced below. When one pushes the hold button on the interface clock, the clock is stopped and a comparison can be made between the displayed time and the time printed on the Teletype.

HR	MIN	SEC
11	6	14
11	6	16
11	6	17
11	6	19
11	6	21
11	6	23
11	6	24
11	6	26
11	6	28
11	6	29
11	6	31
11	6	33
11	6	35
11	6	36
11	6	38

[8/] L_1 is the A-weighted sound level which is exceeded 1 percent of the time. L_{10} is the A-weighted sound level which is exceeded 10 percent of the time. Both L_1 and L_{10} are often used to represent the higher-level, shorter-duration sounds. L_{50} is the A-weighted sound level which is exceeded 50 percent of the time. L_{eq} is the energy equivalent A-weighted sound level.

'BAD' DIGITALLY CONTROLLED SOUND LEVEL SURVEY

PAGE 1 TAPE # 1 YEAR 1975 (2607 CAL 1975 58 2 23)

	DAY	HR	MN	SEC	1- CHANNEL 1 -1 1%	10%	50%	LEQ	1- CHANNEL 2 -1 1%	10%	50%	LEQ	WIND K	AC PFS	# ERR	# SKIP
1	58	2	51	4	74	69	65	67	80	80	79	79	0	0	0	0
2	58	3	1	4	76	68	65	67	80	79	79	79	0	0	0	0
3	58	3	11	4	75	69	66	67	80	80	79	79	0	0	0	0
(	58	3	12	4 LAST TAPE RECORD) PUT SS2 DOWN.												

A WT BUTTON IN, NORMAL NOISE SIGNAL CONDITIONS.
ENTER ATTENUATOR SETTINGS (BOTH ON ONE LINE)
.01 .01

4	58	3	23	0	75	69	66	67	80	79	79	78	0	0	0	0
(	58	3	24	0 LAST TAPE RECORD) PUT SS1 SS2 DOWN												

HAVE A WT. BUTTON OUT ON BOTH 2607 UNITS.
PUT PISTONPHONE ON CH 1, HAVE METER READ ABOUT 44 DB.
ENTER ATTENUATOR SETTING WHILE PISTONPHONE IS ON.
.3 .3
PUT PISTONPHONE ON CH 2, HAVE METER READ ABOUT 44 DB.
ENTER ATTENUATOR SETTING WHILE PISTONPHONE IS ON.
.3
CH 1 DRIFT = -.10 DB CH 2 DRIFT = -.10 DB
A WT BUTTON IN, NORMAL NOISE SIGNAL CONDITIONS.
ENTER ATTENUATOR SETTINGS (BOTH ON ONE LINE)
.03 .03

An example of the printout from the "TRTA1" program is shown below. The listing prints out the sound pressure level (in decibels re 20 μPa) in each one-third octave band from 12.5 Hz to 40 kHz (depending on which filters are utilized in the particular RTA), the sound level for the selected weighting network and the overall sound pressure level (linear). As discussed in Section 4.1.3.c., the underflow indicates whether or not the value is above or below the RTA baseline, while the overflow, i.e., an indication of RTA saturation, is indicated by the printing out on the listing -- RTA INPUT AMPLIFIER WAS OVERLOADED -- if saturation occurred.

PUT 550 UP TO BRING IN DATA FROM RTA.

REAL - TIME 1/3 OCTAVE ANALYZER TYPE 3347 OUTPUT

BAND	DB	UFLOW	BAND	DB	UFLOW
BASE L.	70.0	1	400.0	97.6	0
12.5	78.8	0	500.0	98.0	0
16.0	85.4	0	630.0	98.2	0
20.0	89.6	0	800.0	98.4	0
25.0	97.8	0	1000.0	99.8	0
31.5	101.6	0	1250.0	95.0	0
40.0	102.0	0	1600.0	93.2	0
50.0	99.4	0	2000.0	93.2	0
63.0	101.0	0	2500.0	90.4	0
80.0	100.0	0	3150.0	88.2	0
100.0	99.2	0	4000.0	84.8	0
125.0	98.0	0	5000.0	80.0	0
160.0	98.4	0	6300.0	77.6	0
200.0	100.4	0	8000.0	71.4	0
250.0	102.0	0	10000.0	70.0	1
315.0	98.2	0	12500.0	70.0	1
WEIGHT.	106.0	0	16000.0	70.0	1
LINEAR	113.6	0	20000.0	70.0	1

4.1.5. Procedure for Loading Programs -- "CHECK", "NOISE, "TCLOK" and "TRTA1"

These programs which are written in conversational FORTRAN utilize one or more subroutines which are written in conversational FORTRAN or assembly language. The most efficient procedure for loading any one of these programs (and their subroutines) from paper tape is to utilize a high speed reader (HSR) format. (Copies of "CHECK", "NOISE", "TCLOK", and "TRTA1" have been supplied to BAD in HSR format.)

To run a high speed reader format tape the following procedure should be followed:

1. load XRAY (executive routine)

2. turn on high speed paper tape reader and place paper tape on reader.

3. enter the following command on the Teletype keyboard.

```
:IO, 1, 2Ø
:EX
```

Execution will then be automatic.

For reference purposes, steps 1 to 10 explain the punching order necessary to create a HSR tape.

1. Punch the following step 1 commands on paper tape:

```
line feed :QU,CF carriage return
line feed :EX    carriage return
line feed GC     carriage return
```

2. Copy-punch the FORTRAN main program.

3. If there are any FORTRAN subroutines, continue in to step 4. If there are no FORTRAN subroutines go to step 6.

4. Punch the following command on the paper tape:

   line feed GC carriage return

5. Copy-punch the FORTRAN subroutine. Repeat steps 4 and 5 for all FORTRAN subroutines.

6. Punch the following commands on the paper tape:

```
line feed E       carriage return
line feed :QU,CFR carriage return
```

7. If there are one or more assembly language subroutines, skip to step 9. If there are no assembly language subroutines proceed to step 8.

8. Punch the following command on the paper tape:

   line feed :EX carriage return

   Skip to step 11.

9. Punch the following command on the paper tape:

   line feed :AL carriage return

10. Copy-punch the assembly language subroutines on the paper tape with execution addresses of X'78'. The last subroutine should transfer to X'40'.

11. Punch the following command on the paper tape:

    line feed G carriage return

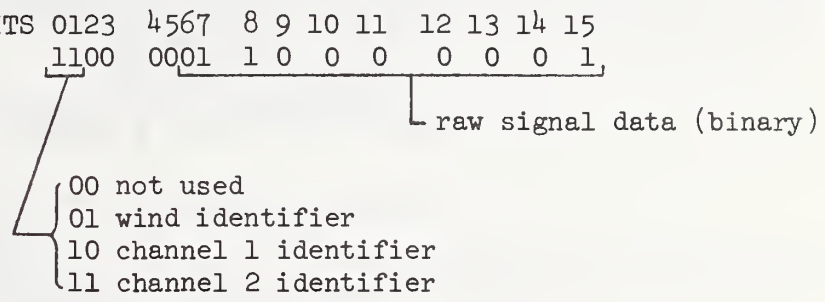
To load the "CHECK" and "NOISE" programs from a special magnetic tape the reader is referred to the discussion of subroutine "STEM1" (Section 4.1.2.a.).

4.1.6. Interface Software Commands

The Interface Software Command (Table 1) lists all the direct input (DIN) and direct output (DOT) commands that affect the community noise interface and the real time analyzer (RTA). Examples for the 16-bit words are given as an aid in understanding the usage of the commands. This table is mainly for reference because these commands are already used in the software drivers supplied with the community noise and real time analyzer programs.

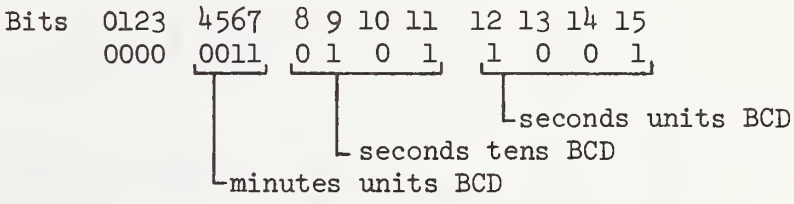
Table 1. Interface Software Commands

Raytheon 704 Command	Effect or Results
DOT 15,6	send master reset to interface (resets PROM counter, and clears FIFO memory)
DIN 15,2	sends 16 bits of data from interface to CPU* accumulator

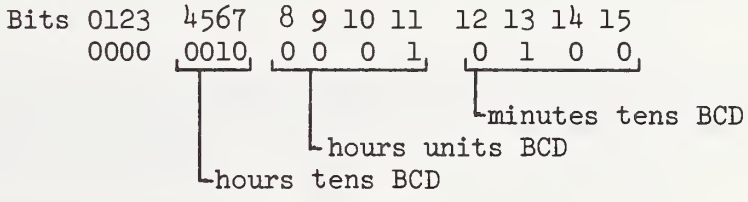


The above example has channel 2 raw data equal to 385.

DOT 15,2	sends 16 bits of data from CPU accumulator to 4 digit hexidecimal display. (Use software to convert binary numbers to decimal BCD** numbers for ordinary base 10 displays)
DIN 15,3	send 16 bits of clock data to CPU accumulator



DIN 15,4	send 16 bits of clock data to CPU accumulator

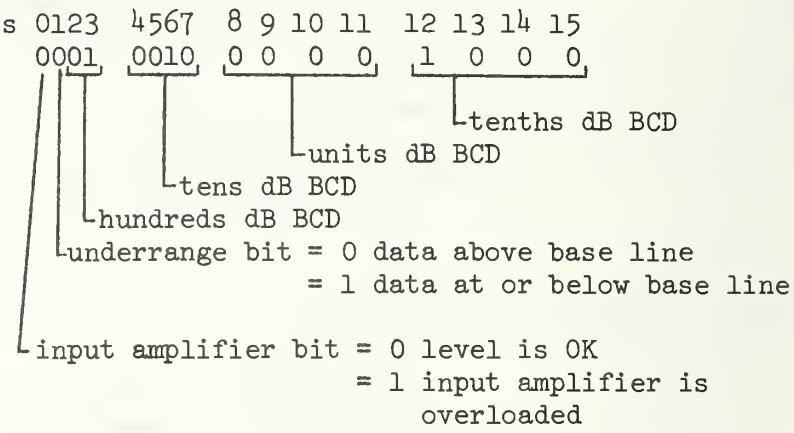


The above time of day example = 21:43:59.

DOT 6,1	send data request to real time analyzer (RTA)

Table 1 Interface Software Commands (Con't)

Raytheon 704 Command	Effect or Results
DIN 6,1	send 16 bits of RTA data to CPU accumulator



The above example shows 120.8 dB for the particular channel digitized by the RTA

* CPU stands for Central Processing Unit
**BCD stands for Binary Coded Decimal

4.2. FORTRAN Software for the Analysis of Community Noise Data

The analysis of the community noise tapes produced by the minicomputer-based data acquisition system is most satisfactorily performed on a large computer system with a high speed printer and digital plotter capability. The main programs and subroutines discussed in this section were specifically written for the Univac 1108 computer system at the National Bureau of Standards but can be adopted for use in almost any large scale system. The software described herein utilizes three utility packages available on the Univac 1108 system, "NTRAN", "AITINT", and "GDS" plotting package, for which detailed documentation was supplied to BAD as part of the operating system.

4.2.1. Nine-track Tape Translation Program

a. Purpose

This program is designed to translate nine-track magnetic tapes written by the 704 Raytheon computer using the Raytheon FORTRAN unformatted write statement. (See program listings in Appendix D.)

In addition to translation of the nine-track tapes, the program also bins the data into selected sound pressure levels with a choice of bin size and range, calculates L_1, L_{10}, L_{50}, L_{90}, L_{99} and L_{eq} and lists the results on a line printer. L_j is that level which is exceeded j percent of the time while L_{eq} is the equivalent sound level averaged on an energy basis.

$$L_{eq} = 10 \log_{10} \left(\frac{\sum_{i=1}^{n} 10^{SPL(i/10)}}{N} \right)$$

where SPL(i) are the sound pressure levels
and N is the total number of data points.

If desired all or part of any block of data can be printed for examination.

b. Control Cards

There are a total of three control cards required which allow (a) choice of bin size and range, (b) choice of the portion of the tape to analyze and (c) choice of data to be displayed.

Card 1: IBEG, IEND, IDEL

Format is free, i.e., numbers separated by commas. All formats are of this type unless stated otherwise.

IBEG is the lower bound of range.

IEND is the upper bound of range.

IDEL is the step size within the range,
e.g., 40, 80, 2 means that the data will be binned
in 20 bins from 40 to 80 dB in 2 dB bins.

Card 2: DAYST, HRST, MINST, DAYFIN, HRFIN, MINFIN

First three are the starting day, hour, minute.
Second three are the finishing day, hour, minute.
e.g. 209, 19, 1, 209, 19, 2 will result in two blocks
of data being read into memory.

Card 3: ICTRL

Legal values are:

0 No listing of the blocks of data.
1 List each control section (1201-1224)
2 List entire 1224 word block

Caution: Use option 2 with care as large amounts of output can be generated.

Note: Each block represents 1 minute of data taken every 0.1 second for two microphones. These 1200 data points are the first 1200 words of the block. In addition there is a 24 word control section at the end of each block giving information pertinent to the block. The output of the program is self explanatory with the exception that each time a read problem occurs (a parity error) -3 is printed and that block is ignored. Also if control word 1207 in the control section is non zero that block is skipped.

c. Subroutines

The program utilizes 3 subroutines for reading, translating and interpolating the data.

1) Reading of the nine-track tapes is accomplished by a routine called RDUNPK which in turn calls the Univac routine NTRAN (described below). RDUNPK (M,N) contains two transfer arguments (M,N). M is the unit number assigned to the tape at execution. N is the status word described in the NTRAN guide. This routine uses NTRAN to read blocks of integers from tape (see below). The 16 digit Raytheon integers are unpacked from the 36 digit Univac words. (The Raytheon integers enter the Univac memory sequentially in 8 bit "bytes" two bytes per integer.) This subroutine is listed in Appendix D under the Nine-track Tape Translation Program.

2) NTRAN is a Univac 1108 Fortran library routine used to read blocks of data from tape or drum.

3) AITINT - is an interpolation program available in the Univac 1108 Math-Pack Library. This program is included with the main program on cards but can be accessed from the fastran drum system at execution time if desired.

4.2.2. Calcomp Plot Program

a. Purpose

Given a specified starting day and hour, this program searches the tape for the starting block; then it calculates L_1, L_{10}, L_{50}, L_{90}, L_{99} and L_{eq} for a specified number of hours (minimum two hours). The six quantities are plotted as a function of time on a simple graph. Each plot contains the data points connected by solid straight lines. Up to a maximum of 100 hours of data can be plotted.

b. Control Cards

Card 1: IBERG, IEND, IDEL

Same interpretation as in Section 4.2.1.a.

Card 2: DAYST, HRST, NHR
Format I6 and Right Justified

These are the starting day and hour. Calculation begins with first good data block in the specified hour and continues for NHR number of hours.

c. Subroutines

This program utilizes RDUNPK, NTRAN, and AITINT which are described in Section 4.1.1. In addition the graphical display system (GDS) subroutines are used. These include: GDLILI, NODLIB, NODLIL, TITLEB, TITLEL, TITLEG, PSLILI, SLLILI, NXTFRM, and GDSEND.

5. APPENDIX A. PROGRAM LISTINGS AND FLOW CHARTS FOR CONVERSATIONAL FORTRAN CALIBRATION AND COMMUNITY NOISE PROGRAMS.

Appendix A contains flow charts for the "CHECK" and "NOISE" programs. It also contains program listings for the "CHECK", "TOGLE", "ORTH", and "NOISE" programs.

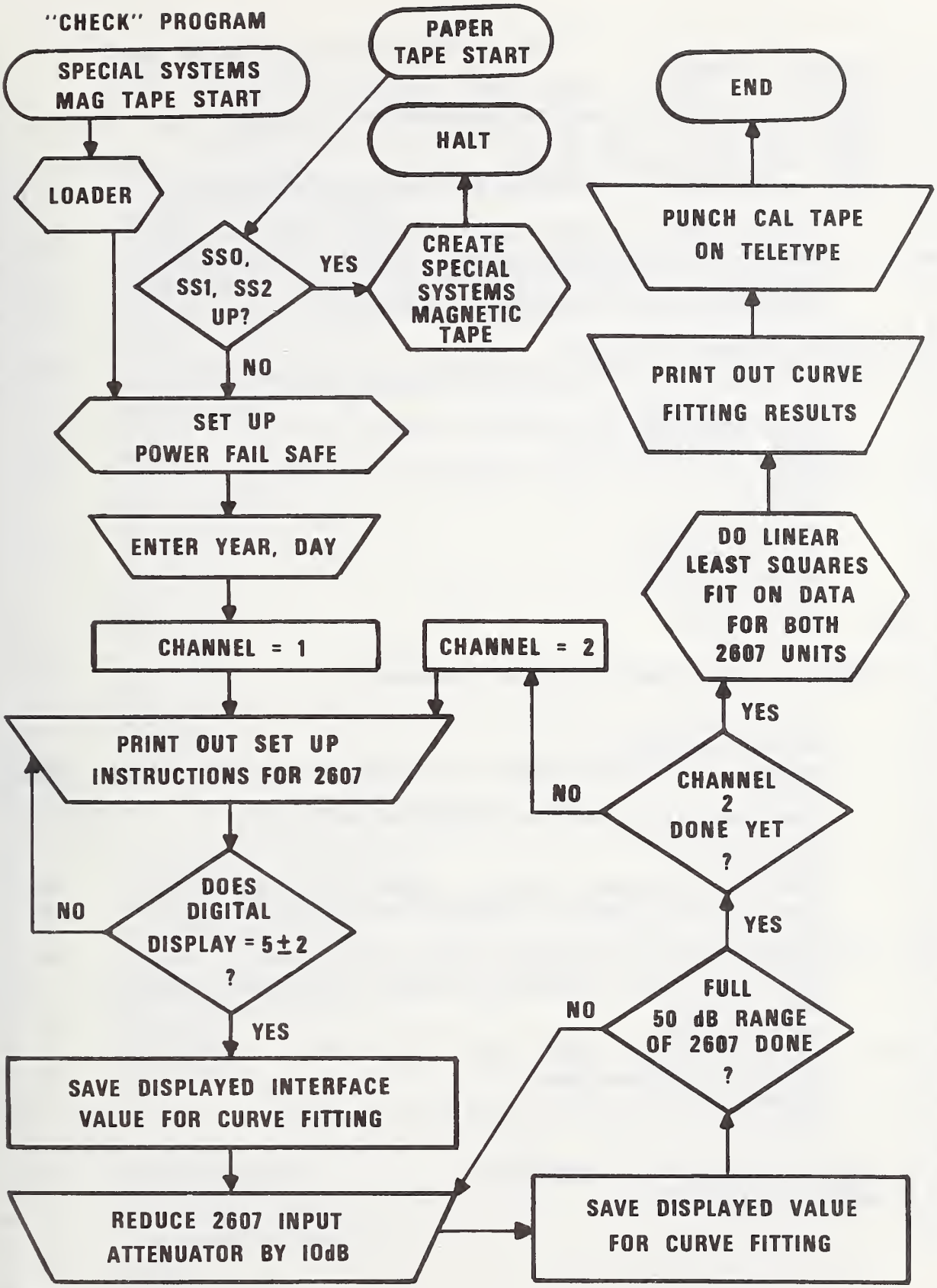


Figure A-1. Flow diagram for "CHECK" program.

```
 1 C 1/3/75:1 'CHECK'
 2 C PROGRAM IS USED TO CHECK THE LINEARITY OF THE
 3 C 2607 ADC INTERFACE SYSTEM AND PRODUCE A PAPER
 4 C TAPE CONTAINING THE DATE, TIME, AND THE FACTOR
 5 C RELATING ADC VALUES TO DB.
 6 C THIS PAPER TAPE IS READ BY THE 'NOISE' PROGRAM.
 7 C
 8       LOCATE STEM1,X'2730',DATA,X'2800',POWF5,X'2890'
 9       LOCATE INTL,X'29E0',LEADR,X'2A20',CLOK,X'2A70'
10       DIMENSION JXX(2,7),XX(2,7),Y(7),X(7),JY(7),CZ(2)
11       DIMENSION YC1(7),DBER1(7),DBER2(7)
12 C
13 C SET UP POWER FAIL SAFE
14 C STEM1 ALLOWS CREATION OF SPECIAL SYSTEM MAG TAPE
15 C
16       CALL STEM1(0)
17       CALL POWF5(NPF)
18       NP=5
19       JZ=0
20       JBOT=5
21       JBOT1=JBOT-2
22       JBOT2=JBOT+2
23       DO 101 J=1,NP
24       JY(J)=J*10
25 101   Y(J)=JY(J)
26 C
27 C CHECK IF ANY POWER FAILURES OCCURED
28 C
29       IF(NPF)3,3,2
30 2     WRITE(3,4)NPF
31 4     FORMAT(3/,'THERE WERE',I5,' POWER FAILURES, RELOAD CHECK')
32       PAUSE
33 3     KK=ISWCH(0)+ISWCH(1)+ISWCH(2)+ISWCH(3)+1
34       GO TO(8,5),KK
35 5     WRITE(3,6)
36 6     FORMAT(3/,'PLEASE PUT ALL SENSE SWITCHES DOWN')
37 7     KK=ISWCH(0)+ISWCH(1)+ISWCH(2)+ISWCH(3)+1
38       GO TO(8,7),KK
39 8     WRITE(3,9)
40 9     FORMAT(3/,'SET INTERFACE CLOCK TO CORRECT TIME')
41       CALL TOGLE
42       WRITE(3,70)
43 70    FORMAT('ENTER  YEAR  DAY    (ON SAME LINE)')
44       READ(3,71)JYEAR,JDAY
45 71    FORMAT(FF)
46       JCH=1
47 C
48 C SET UP LINKAGE TO INTERFACE
49 C ENABLE INTERRUPT 5
50 C
```

```
      CALL DATA(JWIND,JRAW1,JRAW2,JCH)
      CALL INTL(5,2)
C
C BEGIN 2607 - INTERFACE SYSTEM CHECK
C
27    WRITE(3,10)JCH
10    FORMAT(3/,'CHANNEL',I2,' CHECK')
      IF(JCH-1)100,100,102
100   WRITE(3,11)
11    FORMAT('USE 50 MV INTERNAL REFERENCE AS ONLY INPUT TO 2607')
102   WRITE(3,12)
12    FORMAT('SET 2607 INPUT ATTENUATOR TO 3.0 VOLTS')
      IF(JCH-1)103,103,13
103   WRITE(3,95)
95    FORMAT('SET 2607 OUTPUT ATTENUATOR TO X1 POSITION')
      WRITE(3,96)
96    FORMAT('SET 2607 OUTPUT SWITCHES TO LOG AND DC')
13    WRITE(3,14)JBOT
14    FORMAT('ADJUST 2607 VARIABLE GAIN SO DIGITAL DISPLAY =',I3)
      CALL TOGLE
      GO TO(15,16),JCH
15    JR=JRAW1
      A=JR
      JADC1=JR
      XADC1=A
      GO TO 17
16    JR=JRAW2
      A=JR
      JADC2=JR
      XADC2=A
17    JA=A
C
C CONTINUE IF DISPLAY IS NEARLY = JBOT
C
      IF(JA-JBOT1)13,18,18
18    IF(JA-JBOT2)19,19,13
19    DO 24 J=1,NP
      WRITE(3,20)JCH
20    FORMAT('REDUCE 2607 INPUT ATTENUATOR BY 10 DB (CHAN',I2,')')
      CALL TOGLE
      GO TO(21,22),JCH
21    A=JRAW1
      GO TO 23
22    A=JRAW2
C
C SAVE DATA FOR PRINTOUT & LEAST SQUARES FIT
C
23    JXX(JCH,J)=A
      XX(JCH,J)=A
24    CONTINUE
      IF(JCH-2)25,26,26
25    JCH=2
      GO TO 27
26    DO 31 J=1,NP
31    X(J)=XX(1,J)
C
```

```
C DO 1ST ORDER FIT ON CHANNEL 1 DATA
C
      CALL ORTH(X(1),Y(1),CZ(1),YC1(1),ER,1,JRD,1,0,NP,0.)
      DO 32 J=1,NP
32    DBER1(J)=YC1(J)-Y(J)
      FACT1=CZ(2)
      CON1=CZ(1)
      DO 33 J=1,NP
33    X(J)=XX(2,J)
C
C DO 1ST ORDER FIT ON CHANNEL 2 DATA
C
      CALL ORTH(X(1),Y(1),CZ(1),YC1(1),ER,1,JRD,1,0,NP,0.)
      DO 34 J=1,NP
34    DBER2(J)=YC1(J)-Y(J)
      FACT2=CZ(2)
      CON2=CZ(1)
      YDB1=CON1+XADC1*FACT1
      YDB2=CON2+XADC2*FACT2
C
C DETERMINE TIME FROM CLOCK
C
      CALL CLOK(JH,JM,JS)
C
C BEGIN PRINTOUT OF LINEARITY CHECK
C
      WRITE(3,45)
45    FORMAT(15/)
      WRITE(3,46)
46    FORMAT('2607 - INTERFACE NOISE SYSTEM LINEARITY CHECK')
      WRITE(3,47)JYEAR,JDAY
47    FORMAT(/,53X,'YEAR    =',I5,/,53X,'DAY     =',I5)
      WRITE(3,48)JH,JM
48    FORMAT(53X,'HOUR    =',I5,/,53X,'MINUTE =',I5)
      WRITE(3,49)
49    FORMAT(6X,'2607',4X,'CH1',5X,'CH2',5X,'CH1',5X,'CH2')
      WRITE(3,50)
50    FORMAT(7X,'DB',5X,'ADC',5X,'ADC',5X,'DB',6X,'DB')
      WRITE(3,51)
51    FORMAT(13X,'VALUE',3X,'VALUE',4X,'DIFF',4X,'DIFF',/)
      WRITE(3,110)JZ,JADC1,JADC2,YDB1,YDB2
110   FORMAT(4X,I5,3X,I5,3X,I5,F9.2,F8.2,'    (0 DB VALUE NOT USED)')
      DO 52 J=1,NP
52    WRITE(3,53)JY(J),JXX(1,J),JXX(2,J),DBER1(J),DBER2(J)
53    FORMAT(4X,I5,3X,I5,3X,I5,F9.2,F8.2)
      WRITE(3,54)FACT1
54    FORMAT(3/,6X,'CH1 DB = CONSTANT +',E13.5,2X,'* (ADC VALUE)')
      WRITE(3,55)FACT2
55    FORMAT(6X,'CH2 DB = CONSTANT +',E13.5,2X,'* (ADC VALUE)')
      WRITE(3,40)
40    FORMAT(3/,'TURN ON PUNCH, TOGGLE SENSE SWITCH 0')
80    KK=ISWCH(0)+1
      GO TO(80,81),KK
81    KK=ISWCH(0)+1
      GO TO(82,81),KK
C
```

```
C PUNCH LEADER, PUNCH NUMBERS, PUNCH LEADER
C
82    CALL LEADR
      WRITE(3,41)JYEAR,JDAY,JH,JM,FACT1,FACT2
41    FORMAT(I5,I4,I3,I3,1X,E13.5,1X,E13.5)
      CALL LEADR
      IF(NPF)60,60,2
60    WRITE(3,42)
42    FORMAT('TURN OFF PUNCH, TOGGLE SENSE SWITCH 0 TO RESTART')
      WRITE(3,90)
90    FORMAT('CHECK OUT IS COMPLETED. USE PAPER TAPE IN NOISE')
      WRITE(3,91)
91    FORMAT('PROGRAM IF ABOVE RESULTS ARE SATISFACTORY.',3/)
85    KK=ISWCH(0)+1
      GO TO(85,86),KK
86    KK=ISWCH(0)+1
      GO TO(3,86),KK
      END
```

```
 1 C 12/11/74:1 'TOGLE'
 2 C PRINTS OUT 'TOGGLE SENSE SWITCH Ø MESSAGE'
 3 C AND RETURNS TO MAIN PROGRAM WHEN SENSE
 4 C SWITCH Ø IS TOGGLED. (IE. UP & DOWN)
 5 C
 6       SUBROUTINE TOGLE
 7       WRITE(3,1)
 8 1     FORMAT(5X,'(TOGGLE SENSE SWITCH Ø WHEN ABOVE IS OK)',3/)
 9 2     KK=ISWCH(Ø)+1
10       GO TO(2,3),KK
11 3     KK=ISWCH(Ø)+1
12       GO TO(4,3),KK
13 4     RETURN
         END
```

```
 1 C 11/12/74:8 'ORTH'
 2 C
 3       SUBROUTINE ORTH(X,Y,CZ,YC1,ERROR,NHIGH,JRD,KT,NOR,NP,SG)
 4       DIMENSION CP1(10),CP2(10),CP3(10),CP4(10),CX(10)
 5       DIMENSION PV1(50),PV2(50),SS(50)
 6       DIMENSION CZ(10),X(50),Y(50),YC1(50)
 7 C
 8 C ABOVE DIMENSIONS ALLOW FITTING UP TO
 9 C X**9 ORDER AND UP TO 50 DATA POINTS.
10 C
11 C IN MAIN PROGRAM DIMENSION CZ(10),X(50),Y(50),YC1(50)
12 C
13       BONE=1.
14       ZERO=0.
15 C
16 C X IS AN ARRAY CONTAINING VALUES OF THE INDEPENDENT VARIABLE.
17 C Y IS AN ARRAY CONTAINING VALUES OF THE DEPENDENT VARIABLE.
18 C CZ IS AN ARRAY CONTAINING CURVE FITTING MONOMIAL COEFFICIENTS.
19 C ERROR IS THE STANDARD DEVIATION OF (YOBSERVED - YFIT)
20 C NHIGH IS HIGHEST FITTING ORDER IF SUBROUTINE IS REQUESTED
21 C     TO DO COMPLETE FIT ON ONE CALL. (IE. KT NOT = 0)
22 C JRD IS PRESENT ORDER OF FIT. ON SINGLE PASS MODE (KT = 0)
23 C     SET JRD = 0 ON FIRST PASS.
24 C KT = 0 FOR EXIT ON EACH PASS. KT NOT = 0 IN MULTIPASS MODE.
25 C NOR = 0 FOR ORDINARY ORTHOGONAL FIT. NOR = 1 FOR ODD POWER
26 C     ORTHOGNAL FIT. NOR = 2 FOR EVEN POWER ORTHOGONAL FIT.
27 C NP IS THE NUMBER OF DATA POINTS TO BE FITTED.
28 C SG IS DESIRED ERROR OF FIT TEST. IF KT NOT = 0 THE PROGRAM
29 C     EXITS WHEN EITHER JRD = NHIGH OR WHEN ERROR = SG OR LESS.
30 C YC1 IS AN ARRAY CONTAINING POLYNOMIAL VALUES FOR Y (NO NEED
31 C     FOR USER TO DIRECTLY EVALUATE TOTAL POLYNOMIAL TO
32 C     OBTAIN YFIT)
33 C FOR THE BEST ACCURACY OF FIT THE VALUES OF X SHOULD BE
34 C     BETWEEN -2.0 AND +2.0 (IE. TRANSFORM INDEPENDENT
35 C     VARIABLE TO -2.0 +2.0 RANGE BEFORE CALLING SUB ORTH.)
36 C
37       JJM1=NP
38       AM=NP
39       KORD=NHIGH+1
40       IF(KT)703,704,703
41 703    JRD=0
42 704    IF(JRD-1)400,301,302
43 C
44 C ZEROING COEFFICIENTS
45 C
46 400    DO 54 J=1,KORD
47       CP1(J)=ZERO
48       CP2(J)=ZERO
49       CP3(J)=ZERO
50       CP4(J)=ZERO
51       CZ(J)=ZERO
52 54     CX(J)=ZERO
53       S=ZERO
```

```
C
C PART 1 FINDING P OF ORDER ZERO
C
      CP3(1)=BONE
      S1=ZERO
      DO 56 J=1,JJM1
      IF(NOR-1)500,501,500
501   PV1(J)=X(J)
      Z=Y(J)*X(J)
      GO TO 502
500   PV1(J)=BONE
      WN=AM
      Z=Y(J)
502   S=S+Z
56    S1=S1+Y(J)*Y(J)
      IF(NOR-1)503,504,503
504   S2=ZERO
      DO 506 J=1,JJM1
506   S2=S2+X(J)*X(J)
      WN=S2
503   B=S/WN
      DO 325 J=1,JJM1
325   YC1(J)=B
      CX(1)=B
      U=S1-B*B*WN
      Z=U/AM
      Z=ABS(Z)
      SN=SQRT(Z)
      Z=U/(AM-BONE)
      Z=ABS(Z)
      SPOLY=SQRT(Z)
      Z=SQRT(WN)
      SCO=SPOLY/Z
      JRD=1
      JRDM1=0
      JGO=1
      GO TO 100
C
C PART 2 FINDING P OF ORDER 1
C
301   CONTINUE
      S=ZERO
      DO 57 J=1,JJM1
      IF(NOR-1)507,508,630
630   Z=X(J)
      S=S+Z*Z
      GO TO 57
508   Z=X(J)
      S=S+Z*Z*Z*Z
      GO TO 57
507   S=S+X(J)
57    CONTINUE
      AL=S/WN
      S=ZERO
      S1=ZERO
      CP2(1)=CP3(1)
      CP3(2)=BONE
```

```
                CP3(1)=-AL
                WO=WN
                DO 58 J=1,JJM1
                IF(NOR-1)509,510,631
  631           Z=X(J)
                Z=Z*Z-AL
                GO TO 511
  510           Z=X(J)
                Z=Z*Z*Z-AL*Z
                GO TO 511
  509           Z=X(J)-AL
  511           PV2(J)=Z
                S=S+Z*Z
   58           S1=S1+Y(J)*Z
                WN=S
                B=S1/WN
                DO 326 J=1,JJM1
  326           YC1(J)=YC1(J)+B*PV2(J)
                CX(1)=CX(1)+CP3(1)*B
                CX(2)=B
                BT=WN/WO
                U=U-B*B*WN
                Z=U/AM
                Z=ABS(Z)
                SN=SQRT(Z)
                Z=U/(AM-2.)
                Z=ABS(Z)
                SPOLY=SQRT(Z)
                Z=SQRT(WN)
                SCO=SPOLY/Z
                JRD=2
                JRDM1=1
                JGO=2
                GO TO 100
C
C PART 3 FINDING P OF ORDER JRD - 1
C
  302           JRD=JRD+1
                BORD=JRD
                JRDM1=JRD-1
                JRDM2=JRD-2
                S=ZERO
                DO 61 J=1,JJM1
                Z=PV2(J)
                IF(NOR-1)512,513,513
  513           S2=X(J)
                S=S+S2*S2*Z*Z
                GO TO 61
  512           S=S+X(J)*Z*Z
   61           CONTINUE
                AL=S/WN
                S=ZERO
                S2=ZERO
                DO 62 J=1,JRDM2
   62           CP1(J)=CP2(J)
                DO 63 J=1,JRDM1
   63           CP2(J)=CP3(J)
```

```
      DO 308 J=1,JRD
308   CP3(J)=ZERO
      DO 64 J=1,JRDM2
64    CP3(J)=-CP1(J)*BT
      DO 65 J=1,JRDM1
65    CP3(J)=CP3(J)-AL*CP2(J)
      DO 66 J=2,JRD
66    CP3(J)=CP3(J)+CP2(J-1)
      WO=WN
      DO 68 J=1,JJM1
      Z=X(J)
      IF(NOR-1)514,515,515
515   Z=Z*Z
514   S1=-BT*PV1(J)+(Z-AL)*PV2(J)
      S=S+S1*S1
      S2=S2+Y(J)*S1
      PV1(J)=PV2(J)
68    PV2(J)=S1
      WN=S
      B=S2/WN
      DO 327 J=1,JJM1
327   YC1(J)=YC1(J)+B*PV2(J)
      DO 67 J=1,JRD
67    CX(J)=CX(J)+CP3(J)*B
      BT=WN/WO
      U=U-B*B*WN
      Z=U/AM
      Z=ABS(Z)
      SN=SQRT(Z)
      Z=U/(AM-BORD)
      Z=ABS(Z)
      SPOLY=SQRT(Z)
      Z=SQRT(WN)
      SCO=SPOLY/Z
      JGO=3
      GO TO 100
C
C SETTING UP ORDINARY, ODD, OR EVEN ORTHOGONAL COEFFICIENTS
C     AND THEIR MONOMIALS
C
100   S=ZERO
      DO 328 J=1,JJM1
      SS(J)=Y(J)-YC1(J)
328   S=S+SS(J)
      AVYC=S/AM
      S=ZERO
      DO 329 J=1,JJM1
329   S=S+(AVYC-SS(J))*(AVYC-SS(J))
      ERROR=SQRT(S/AM)
      IF(KT)700,621,700
700   IF(JRD-KORD)701,621,621
701   IF(ERROR-SG)621,621,702
702   GO TO (301,302,302),JGO
621   IF(NOR-1)516,517,518
516   JORD=JRD
      DO 519 J=1,JRD
      CP4(J)=CP3(J)
```

```
225 519    CZ(J)=CX(J)
226        GO TO 520
227 517    JORD=JRD*2
228        DO 521 J=1,JORD
229        K=(J+1)/2
230        IF(J-2*(J/2))522,522,521
231 522    CZ(J)=CX(K)
232        CP4(J)=CP3(K)
233 521    CONTINUE
234        GO TO 520
235 518    JORD=JRD*2-1
236        DO 523 J=1,JORD
237        K=(J+1)/2
238        IF(J-2*(J/2))523,523,524
239 524    CZ(J)=CX(K)
240        CP4(J)=CP3(K)
241 523    CONTINUE
242 520    CONTINUE
243        RETURN
           END
```

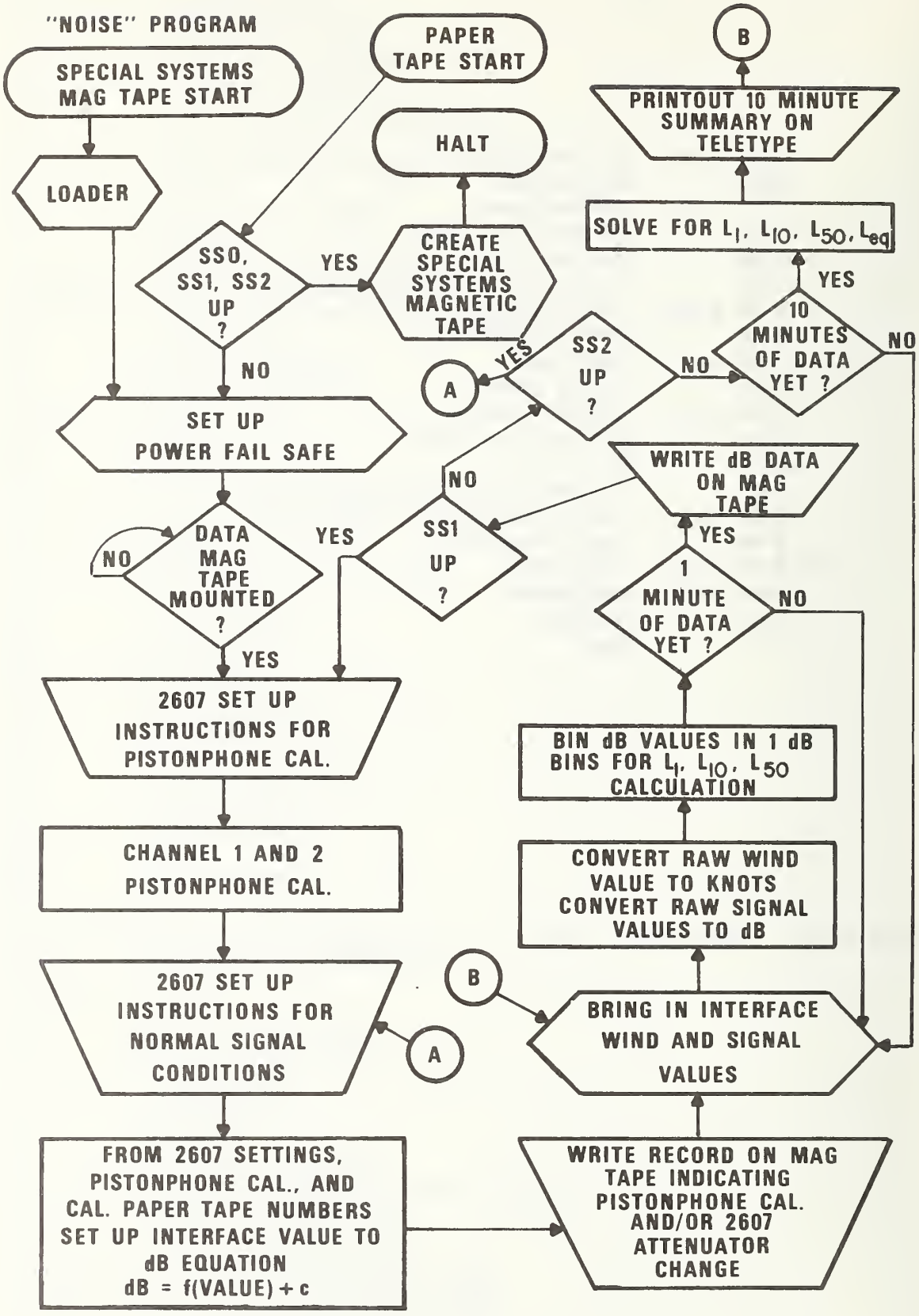


Figure A-2. Flow diagram for "NOISE" program.

```
C 1/23/75:4 'NOISE'
C THE PROGRAM CONDUCTS A TWO CHANNEL COMMUNITY
C NOISE SURVEY BY SAMPLING TWO DATA CHANNELS OF
C ENVIRONMENTAL NOISE AND A OPTIONAL WIND CHANNEL
C AT A 10 HZ RATE. ONCE A MINUTE THE NOISE DATA AND
C THE MAXIMUM WIND VALUE ARE DUMPED ON TO THE MAGNETIC
C TAPE UNIT FOR LATER ANALYSIS ON A UNIVAC 1108 COMPUTER.
C THE PROGRAM WILL PRINTOUT L1, L10, L50, LEQ,
C AND MAXIMUM WIND SPEED EVERY 10 MINUTES ON THE TELETYPE.
C
      LOCATE STEM1,X'2730',DATA,X'2800',LFD,X'2870',POWF5,X'2890'
      LOCATE MULT1,X'291B',MMON,X'299B',DMAMAG,X'29C0'
      LOCATE INTL,X'29E0',DISP,X'2A20',CLOK,X'2A70'
      DIMENSION K(1224),KPA(111),KPB(111)
C
C K(1) TO K(1200) VERY SOON AFTER EACH ELEMENT IS FILLED WITH
C                 AN ADC VALUE (VIA INTERRUPT LEVEL 5) IT IS
C                 CONVERTED TO DB.
C K(1201) = DAY
C K(1202) = HOUR
C K(1203) = MINUTE
C K(1204) = SECOND IMMEDIATELY AFTER RECEIVING & CONVERTING
C           K(1200) VALUE AND JUST BEFORE WRITING ALL 1224
C           NUMBERS ON MAG TAPE.
C K(1205) = NUMBER OF POWER FAILURES IN JUST THE PAST MINUTE.
C K(1206) = PROGRAM VERSION NUMBER
C K(1207) = 0  K ARRAY RECORD ON MAG TAPE CONTAINS REGULAR DB DATA
C         = 1  RECORD ON MAG TAPE INDICATES PISTONPHONE CAL
C         = 2  RECORD ON MAG TAPE INDICATES JUST 2607 ATTENUATOR
C              CHANGE
C K(1208) = MAG TAPE ID NUMBER
C K(1209) = HIGHEST WIND SPEED IN KNOTS FOR THE PAST MINUTE.
C K(1210) = NUMBER OF CHANNEL ID ERRORS GENERATED BY INTERFACE.
C K(1211) = NUMBER OF MISSING DATA VALUES CAUSED BY OVERFLOWING
C           FIFO MEMORY.
C K(1212) = CHANNEL 1 PISTONPHONE VALUE * 10. USING NEXT TO LAST
C           CAL VALUE. (SHOWS CHANNEL 1 SYSTEM DRIFT)
C K(1213) = CHANNEL 2 PISTONPHONE VALUE * 10. USING NEXT TO LAST
C           CAL VALUE. (SHOWS CHANNEL 2 SYSTEM DRIFT)
C K(1214) = YEAR
C K(1215) TO K(1224) = 0 (NOT USED)
      DO 460 J=1215,1224
460   K(J)=0
C
C STEM1 ALLOWS CREATION OF SPECIAL SYSTEM MAG TAPE.
C POWF5 SETS UP POWER FAIL SAFE.
C
```

```
      CALL STEM1(1)
      CALL POWF5(NPF)
      ALGC=1./ALOG(10.)
C
C NFIL + 17 = NUMBER OF LINES PER PAGE IN PRINTOUT.
C PCAL = PISTONPHONE DB VALUE.
C
      NFIL=34
      NTIME=0
      M=0
      N=0
      PCAL=124.
      JPG=1
      WRITE(3,220)
220   FORMAT(3/,'PUT SS1 UP FOR PISTONPHONE CAL. (NOT NOW)')
      WRITE(3,25)
25    FORMAT('PUT SS2 UP TO CHANGE 2607 SETTINGS. (NOT NOW)')
      KK=ISWCH(0)+ISWCH(1)+ISWCH(2)+ISWCH(3)+1
      GO TO(340,341),KK
341   WRITE(3,342)
342   FORMAT(3/,'PLEASE PUT ALL SENSE SWITCHES DOWN.')
343   KK=ISWCH(0)+ISWCH(1)+ISWCH(2)+ISWCH(3)+1
      GO TO(340,343),KK
340   JWAY=0
      WRITE(3,73)
73    FORMAT(2/,'SET INTERFACE CLOCK TO CORRECT TIME.')
      CALL TOGLE
      WRITE(3,138)
138   FORMAT('ENTER  YEAR DAY TAPE#  (3 NUMBERS ON ONE LINE)')
      READ(3,4)JYEAR,JDAY,NTAPE
4     FORMAT(FF)
      K(1208)=NTAPE
      K(1214)=JYEAR
      CALL CLOK(JHR,JMIN,JSEC)
      JHOLD=JHR
      WRITE(3,270)
270   FORMAT(3/,'TURN ON HIGH SPEED PAPER TAPE READER.')
      WRITE(3,271)
271   FORMAT('PUT CAL TAPE (FROM CHECK PROGRAM) IN READER.')
      CALL TOGLE
C
C READ CAL TAPE TO OBTAIN SLOPE OF LINEAR TRANSFER FUNCTION
C      RELATING DB TO ADC VALUE.
C
      READ(4,4)JYRC,JDC,JHC,JMC,FACT1,FACT2
      WRITE(3,272)
272   FORMAT('TURN OFF HIGH SPEED PAPER TAPE READER.')
      WRITE(3,7)
7     FORMAT(3/,'MOUNT DATA TAPE & BRING TO LOAD POINT & ON LINE.')
207   CALL DMAMAG(MAG)
      MAGG=MAG
      IF(MAG-3)207,135,207
135   WRITE(3,240)
240   FORMAT(3/,'DO THE FOLLOWING FOR BOTH 2607 UNITS.')
```

```
102         WRITE(3,241)
103 241     FORMAT(3X,'SET VARIABLE GAIN CONTROL TO CAL POSITION.')
104         WRITE(3,242)
105 242     FORMAT(3X,'SET OUTPUT ATTENUATOR TO X1 POSITION.')
106         WRITE(3,243)
107 243     FORMAT(3X,'A WT. BUTTON SHOULD BE OUT.')
108         WRITE(3,244)
109 244     FORMAT(3X,'50 MV INTERNAL REFERENCE BUTTON SHOULD BE OUT.')
110         WRITE(3,280)
111 280     FORMAT(3X,'SET 2607 OUTPUTS TO LOG, DC, RMS, & .1 SEC.')
112         CALL TOGLE
113         WRITE(3,245)
114 245     FORMAT('PUT PISTONPHONE ON CHANNEL 1.')
115         WRITE(3,246)
116 246     FORMAT('OBTAIN 44 DB METER READING BY ADJUSTING')
117         WRITE(3,248)
118 248     FORMAT(3X,'INPUT ATTENUATOR & SENSITIVITY SCREW.')
119         WRITE(3,247)
120 247     FORMAT('REPEAT ABOVE FOR CHANNEL 2.')
121         CALL TOGLE
122 295     WRITE(3,281)
123 281     FORMAT('HAVE A WT. BUTTON OUT ON BOTH 2607 UNITS.')
124         WRITE(3,250)
125 250     FORMAT('PUT PISTONPHONE ON CH 1, HAVE METER READ ABOUT 44 DB.')
126         CALL DATA(JXX,JRAW1,JRAW2,3)
127         CALL INTL(5,2)
128         WRITE(3,251)
129 251     FORMAT('ENTER ATTENUATOR SETTING WHILE PISTONPHONE IS ON.')
130         READ(3,4)PSET1
131         AJR1=JRAW1
132         K(1207)=1
133         OCON1=CON1
134         CON1=PCAL-FACT1*AJR1
135         DB=CON1+FACT1*AJR1
136         KKDA=DB*10.
137         CALL DISP(KKDA,0)
138         WRITE(3,252)
139 252     FORMAT('PUT PISTONPHONE ON CH 2, HAVE METER READ ABOUT 44 DB.')
140         WRITE(3,253)
141 253     FORMAT('ENTER ATTENUATOR SETTING WHILE PISTONPHONE IS ON.')
142         READ(3,4)PSET2
143         AJR2=JRAW2
144         OCON2=CON2
145         CON2=PCAL-FACT2*AJR2
146         DB=CON2+FACT2*AJR2
147         KKDB=DB*10.
148         CALL DISP(KKDB,0)
149         K(1212)=KKDA
150         K(1213)=KKDB
151         IF(JWAY)262,263,262
152 262     DBLD1=OCON1+FACT1*AJR1
153         DBLD2=OCON2+FACT2*AJR2
```

```
154 C
155 C CALCULATE SYSTEM DRIFT IN DB FROM PREVIOUS PISTONPHONE READING.
156 C
157       DRFT1=DBLD1-PCAL
158       DRFT2=DBLD2-PCAL
159       K(1212)=DBLD1*10.
160       K(1213)=DBLD2*10.
161       M=M+8
162       WRITE(3,265)DRFT1,DRFT2
163 265   FORMAT('CH 1 DRIFT =',F8.2,' DB   CH 2 DRIFT =',F8.2,' DB')
164 263   JWAY=1
165 311   WRITE(3,290)
166 290   FORMAT('A WT BUTTON IN, NORMAL NOISE SIGNAL CONDITIONS.')
167       WRITE(3,291)
168 291   FORMAT('ENTER ATTENUATOR SETTINGS (BOTH ON ONE LINE)')
169       READ(3,4)SET1,SET2
170       NTIME=NTIME+1
171       IF(K(1207))355,356,355
172 356   K(1207)=2
173 355   M=M+4
174       DW1=20.*(ALOG(PSET1/SET1))*ALGC
175       DW2=20.*(ALOG(PSET2/SET2))*ALGC
176       ADW1=ABS(DW1)+.5
177       ADW2=ABS(DW2)+.5
178       JDW1=ADW1
179       JDW2=ADW2
180       IF(DW1)450,451,451
181 450   JDW1=-JDW1
182 451   IF(DW2)452,453,453
183 452   JDW2=-JDW2
184 453   DBDW1=JDW1
185       DBDW2=JDW2
186 C
187 C CALCULATE INTERCEPT OF LINEAR TRANSFER FUNCTION RELATING DB
188 C       TO ADC VALUE.
189 C
190       CCON1=CON1-DBDW1
191       CCON2=CON2-DBDW2
192       CALL INTL(5,1)
193       DO 360 J=1,1200
194 360   K(J)=0
195       CALL CLOK(JJHR,JJMN,JJSEC)
196       JJDAY=JDAY
197       IF(JHOLD-23)422,421,422
198 421   IF(JJHR)422,423,422
199 423   JJDAY=JJDAY+1
200 422   K(1201)=JJDAY
201       K(1202)=JJHR
202       K(1203)=JJMN
203       K(1204)=JJSEC
204 C
205 C WRITE SPECIAL RECORD ON MAG TAPE INDICATING NEW PISTONPHONE
206 C     READINGS OR 2607 ATTENUATOR CHANGES.
207 C
```

```
208         WRITE(9)K
209         JCNT=0
210 C
211 C START TWO CHANNEL DATA COLLECTION.
212 C
213         CALL MULT1(K(1),JWIND,JCNT,JDONE,JEROR,JSKIP)
214         CALL INTL(5,2)
215         IF(NTIME-1)33,33,400
216 33      DO 351 J=1,3
217         CALL LFD
218 351     CONTINUE
219         WRITE(3,30)
220 30      FORMAT('--')
221         DO 352 J=1,5
222         CALL LFD
223 352     CONTINUE
224         WRITE(3,22)
225 22      FORMAT(5H'BAD',' DIGITALLY CONTROLLED SOUND LEVEL SURVEY')
226         CALL LFD
227         CALL LFD
228         WRITE(3,21)JPG,NTAPE,JYEAR,JYRC,JDC,JHC,JMC
229 21      FORMAT('PAGE',I4,3X,'TAPE # ',I5,3X,'YEAR ',I5,3X,'(2607 CAL'5
230        14,I3,I3,')')
231         N=1
232         M=1
233         KSW=0
234 26      DO 50 J=1,111
235         KPA(J)=0
236 50      KPB(J)=0
237         K1A=0
238         K10A=0
239         K50A=0
240         K1B=0
241         K10B=0
242         K50B=0
243         K(1212)=0
244         K(1213)=0
245         K(1206)=3
246         K(1207)=0
247         JWM1=-30000
248         NPF1=0
249         JER=0
250         JSKP=0
251         ALQA=0.
252         ALQB=0.
253         DO 20 J1=1,10
254         IF(KSW)316,315,316
255 315     IF(J1-2)316,317,316
```

```
256 C
257 C THE HEADING IS LONG AND TO ALLOW ENOUGH TIME FOR ADC VALUE TO
258 C     DB CONVERSION (CONVERSION HALTS DURING TTY PRINTOUT) THE
259 C     SECOND HALF OF THE HEADING IS PRINTED OUT DURING THE SECOND
260 C     MINUTE OF DATA COLLECTION. BRINGING IN ADC VALUES FROM FIFO
261 C     MEMORY (VIA INTERRUPT LEVEL 5) IS HALTED ONLY WHEN THE MAG
262 C     TAPE IS WRITING (ABOUT .5 SECOND) OR WHEN A NEW PISTONPHONE
263 C     CALIBRATION OR 2607 ATTENUATOR CHANGE IS REQUESTED BY SS1
264 C     OR SS2.
265 C
266 317   WRITE(3,23)
267 23    FORMAT(2/,20X,'1- CHANNEL 1 -1    1- CHANNEL 2 -1  WIND  AC  ##
268      1')
269       WRITE(3,90)
270 90    FORMAT(5X,'DAY HR MN SEC   1% 10% 50% LEQ    1% 10% 50% LEQ  K
271      1,' PFS ERR SKIP',/)
272       KSW=1
273 316   JWMAX=30000
274 C
275 C DO LOOP WHERE ADC VALUES ARE CONVERTED TO DB AND WHERE DB VALUES
276 C     ARE BINNED IN 1 DB STEPS.
277 C
278       DO 8 J=1,1200
279 80    IF(JCNT-J)80,147,147
280 147   IF(JCNT-1200)100,41,41
281 41    CALL INTL(5,1)
282 100   ANVAL=K(J)
283       IF(JWMAX-JWIND)125,125,126
284 126   JWMAX=JWIND
285 125   AJ=J
286       A=J/2
287       B=AJ/2.
288       IF(A-B)10,11,10
289 C
290 C CHANNEL 1 ADC DATA CONVERSION TO DB
291 C
292 10    DBA=CCON1+FACT1*ANVAL
293       KDBA=DBA*10.
294       CALL DISP(KDBA,1)
295       ALQA=ALQA+10.**(DBA/10.)
296       KP1=DBA+.5
297       IF(KP1-20)8,43,110
298 110   IF(KP1-130)43,43,8
299 C
300 C BINNING CHANNEL 1 DB VALUES IN 1 DB STEPS.
301 C
302 43    KPA(KP1-19)=KPA(KP1-19)+1
303       GO TO 8
304 C
305 C CHANNEL 2 ADC DATA CONVERSION TO DB
306 C
307 11    DBA=CCON2+FACT2*ANVAL
308       KDBA=DBA*10.
309       CALL DISP(KDBA,2)
```

```
310         ALQB=ALQB+10.**(DBA/10.)
311         KP1=DBA+.5
312         IF(KP1-20)8,62,111
313  111    IF(KP1-130)62,62,8
314  C
315  C BINNING CHANNEL 2 DB VALUES IN 1 DB STEPS.
316  C
317  62     KPB(KP1-19)=KPB(KP1-19)+1
318  8      K(J)=DBA*10.
319          CALL CLOK(JHR,JMIN,JSEC)
320  120    IF(JHOLD-23)122,123,122
321  123    IF(JHR)122,124,122
322  124    JDAY=JDAY+1
323  122    JHOLD=JHR
324          K(1201)=JDAY
325          K(1202)=JHR
326          K(1203)=JMIN
327          K(1204)=JSEC
328          K(1205)=NPF
329          K(1210)=JEROR
330          JER=JER+JEROR
331          K(1211)=JSKIP
332          AJWMX=JWMAX
333          VOLT=-.00821541*AJWMX+4.22168
334          JWM=12.5815*VOLT+3.53
335          IF(JWM-5)150,151,151
336  150    JWM=0
337  151    K(1209)=JWM
338          IF(JWM1-JWM)127,128,128
339  127    JWM1=JWM
340  C
341  C WRITE DATA FOR CURRENT MINUTE ON MAG TAPE.
342  C
343  128    WRITE(9)K
344          NPF1=NPF1+NPF
345          JSKP=JSKP+JSKIP
346          JSKIP=0
347          JEROR=0
348          NPF=0
349          JCNT=0
350          KK=ISWCH(1)+ISWCH(2)+1
351          GO TO(160,320),KK
352  320    IF(J1-10)156,20,20
353  C
354  C HALT DATA COLLECTION IF MAG TAPE IS FULL. (EOT)
355  C
356  160    CALL DMAMAG(MAG)
357          IF(MAG-17)401,402,402
358  401    CALL INTL(5,2)
359  20     CONTINUE
360  28     N1PA=60
361          N10PA=600
362          N50PA=3000
363          N1PB=60
364          N10PB=600
```

```
365       N50PB=3000
366 C
367 C COMPUTE 10 MINUTE VERSION OF LEQ.
368 C
369       LEQA=10.*ALOG(ALQA/6000.)/ALOG(10.)
370       LEQB=10.*ALOG(ALQB/6000.)/ALOG(10.)
371       JUMA=0
372       JUMB=0
373 C
374 C DO LOOP TO ADD UP DB BINS AND DETERMINE 1%, 10%, 50% LEVELS.
375 C
376       DO 60 J2=1,111
377       J3=112-J2
378       JUMA=JUMA+KPA(J3)
379       JUMB=JUMB+KPB(J3)
380       IF(JUMA-N1PA)45,46,46
381 46    N1PA=30000
382       K1A=J3+19
383 45    IF(JUMA-N10PA)47,48,48
384 48    N10PA=30000
385       K10A=J3+19
386 47    IF(JUMA-N50PA)52,51,51
387 51    N50PA=30000
388       K50A=J3+19
389 52    IF(JUMB-N1PB)54,55,55
390 55    N1PB=30000
391       K1B=J3+19
392 54    IF(JUMB-N10PB)56,57,57
393 57    N10PB=30000
394       K10B=J3+19
395 56    IF(JUMB-N50PB)60,59,59
396 59    N50PB=30000
397       K50B=J3+19
398 60    CONTINUE
399 C
400 C OUTPUT DATA TO TTY EVERY 10 MINUTES.
401 C
402 76    WRITE(3,35)N,JDAY,JHR,JMIN,JSEC,K1A,K10A,K50A,LEQA,K1B,K10B,K
403      1EQB,JWM1,NPF1,JER,JSKP
404 35    FORMAT(I4,I4,I3,I3,I3,1X,I5,I4,I4,I4,1X,I5,I4,I4,I4,1X,I5,I4,
405     1)
406 156   KK=ISWCH(1)+1
407       GO TO(301,302),KK
408 302   WRITE(3,303)JDAY,JHR,JMIN,JSEC
409 303   FORMAT('    (',I4,I3,I3,I3,' LAST TAPE RECORD) PUT SS1 SS2 DOWN
410 304   KK=ISWCH(1)+ISWCH(2)+1
411       GO TO(295,304),KK
412 301   KK=ISWCH(2)+1
413       GO TO(307,308),KK
414 308   WRITE(3,309)JDAY,JHR,JMIN,JSEC
415 309   FORMAT('    (',I4,I3,I3,I3,' LAST TAPE RECORD) PUT SS2 DOWN.')
416 310   KK=ISWCH(2)+1
417       GO TO(311,310),KK
```

```
418 307   N=N+1
419       M=M+1
420 400   MORE=M-NFIL
421       MOR2=17-(M-NFIL)
422 C
423 C     IF CURRENT PAGE IS FULL START NEW PAGE.
424 C
425       IF(MORE)26,31,31
426 31    IF(MOR2)332,332,330
427 330   DO 331 J=1,MOR2
428       CALL LFD
429 331   CONTINUE
430 332   WRITE(3,350)
431 350   FORMAT('-----')
432       JPG=JPG+1
433       GO TO 33
434 402   WRITE(3,403)JDAY,JHR,JMIN,JSEC
435 403   FORMAT('    (',I4,I3,I3,I3,' LAST TAPE RECORD) END OF TAPE.')
436       WRITE(3,404)
437 404   FORMAT('MOUNT NEW DATA TAPE, RESET CPU, GO TO ',10HX'80', RUN)
438       WRITE(3,410)
439 410   FORMAT('ON TTY DO FOLLOWING',/,'LINE FEED G CARRIAGE RETURN',3
440       REWIND 14
441 405   GO TO 405
          END
```

6. APPENDIX B. PROGRAM LISTINGS FOR ASSEMBLY LANGUAGE SUBROUTINES

Appendix B contains program listings for "STEM1", "DATA", "LFD", "POWF5", "MULTI1", "MMON", "DMAMAG", "INTL", "DISP", "LEADR", and "CLOK" programs.

```
                1 ' 11/11/74:5 'STEM1'
                2         ORIG  X'2730'
                3 * THIS PROGRAM WHEN CALLED BY C. FORTRAN MAIN
                4 * PROGRAM DUMPS ALL OF CORE ONTO MAG TAPE
                5 * SO THAT THE SYSTEM CAN BE STARTED UP BY:
                6 * 1) MOUNTING SPECIAL SYSTEM TAPE ON DMA DRIVE
                7 * 2) PUSH RESET ON CPU
                8 * 3) PUSH MAG TAPE LOAD BUTTON N TIMES
                9 *          WHERE N = 1 + 2 X NUMBER OF
               10 *                    PROGRAMS AHEAD OF
               11 *                    DESIRED PROGRAM
               12 * 4) PUSH RUN ON CPU
               13 * 5) PUSH RUN ON CPU
               14 *
               15 * SS0, SS1, SS2 MUST BE UP WHEN THIS
               16 * PROGRAM IS CALLED TO CREATE SPECIAL
               17 * MAGNETIC TAPE.
               18 *
               19 * THIS PROGRAM EXAMINES THE PEAT TABLE
               20 * TO CHECK THE SYSTEM DIRECTIVE INPUT
               21 * ASSIGNMENT. (LU 1) IF IT IS ASSIGNED
               22 * TO THE HIGH SPEED READER THE PROGRAM
               23 * CHANGES THE ASSIGNMENT TO THE TTY.
               24 *
               25 * IN FORTRAN PROGRAM USE:
               26 *       CALL STEM1(NUM)
               27 * WHERE:
               28 *       NUM = NUMBER OF PROGRAMS
               29 *             ALREADY ON MAG TAPE
               30 *
2730 0082      31 STEM    DATA  X'82'
2731 0000      32 NUM     DATA  0
2732 0080      33         SMB   0
2733 9059      34         LDX   X'59'  ADDRESS OF PEAT TABLE
2734 8801      35         LDW * 1      SYSTEM DIRECTIVE UNIT #
2735 F7E5      36         CMW   X1400
2736 0860      37         SEQ          IS IT HIGH SPEED READER?
2737 173A      38         JMP   CONT2  NO
2738 87E6      39         LDW   XD00   YES, CHANGE TO TTY
2739 7801      40         STW * 1
273A 08C0      41 CONT2   SS0
273B 173D      42         JMP   SS1
273C 17CA      43         JMP   RET
273D 08D0      44 SS1     SS1
273E 1740      45         JMP   SS2
273F 17CA      46         JMP   RET
2740 08E0      47 SS2     SS2
2741 1743      48         JMP   CONT
2742 17CA      49         JMP   RET
2743 0080      50 CONT    SMB   0
2744 805C      51         LDW   X'5C'
2745 77EB      52         STW   CORE   LAST CORE LOCATION
2746 A7E2      53         ADD   D1
2747 77D4      54         STW   CSIZE  TOTAL # OF WORDS IN CORE
2748 0220      55 STAT4   DIN   2,0    MAG TAPE STATUS
2749 0A1D      56         SLL   13
274A 0820      57         SAM
```

```
274B 179A    58          JMP   RING    NEED WRITE RING
274C 97EB    59          LDX   CORE
274D 8800    60          LDW * 0
274E 77ED    61          STW   SF      SAVE CONT. OF CORE END
274F 0501    62          DXS   1
2750 8800    63          LDW * 0
2751 77EC    64          STW   SE      SAVE CONT. CORE END -1
2752 9731    65          LDX   NUM
2753 8800    66          LDW * 0       # PROGS ALREADY ON TAPE
2754 0A11    67          SLL   1       X2
2755 77EA    68          STW   NUMP
2756 0800    69          SAZ           IS TAPE EMPTY?
2757 1759    70          JMP   SKIP    NO
2758 176F    71          JMP   CONT1   YES
2759 0220    72 SKIP     DIN   2,0     MAG TAPE STATUS
275A 0810    73          SAP           CONTROLLER BUSY?
275B 1759    74          JMP   SKIP    YES, WAIT
275C 0A11    75          SLL   1
275D 0810    76          SAP           TAPE UNIT BUSY?
275E 1759    77          JMP   SKIP    YES, WAIT
275F 87EB    78          LDW   CORE
2760 0321    79          DOT   2,1     START ADDRESS = END CORE
2761 87E2    80          LDW   D1      WON'T USE DATA
2762 0326    81          DOT   2,6     READ RECORD
2763 0220    82 STAT3    DIN   2,0     MAG TAPE STATUS
2764 0A17    83          SLL   7
2765 0820    84          SAM           DONE READING RECORD?
2766 1763    85          JMP   STAT3   NO, WAIT
2767 E7E4    86          AND   X0180   KEEP BITS 14,15
2768 0800    87          SAZ
2769 17B2    88          JMP   ERR     RATE OR TAPE ERROR
276A 87EA    89          LDW   NUMP
276B B7E2    90          SUB   D1
276C 77EA    91          STW   NUMP
276D 0800    92          SAZ
276E 1759    93          JMP   SKIP    MORE TO SKIP
276F 87E3    94 CONT1    LDW   D9
2770 77E7    95          STW   WC      SET # OF WORDS = 9
2771 87E9    96          LDW   LLOAD   SET LOADER START
2772 77E8    97          STW   MEM     ADDRESS
2773 277E    98          JSX   MAG     GO WRITE LOADER ON TAPE
2774 0100    99          CLR
2775 77E8   100          STW   MEM     SET START ADDRESS = 0
2776 87D4   101          LDW   CSIZE
2777 77E7   102          STW   WC      SET # WORDS = ALL OF CORE
2778 27D5   103          JSX   CSUM    CALCULATE CHECK SUM
2779 97EB   104          LDX   CORE
277A 7800   105          STW * 0       STUFF C SUM IN CORE END
277B 277E   106          JSX   MAG     WRITE ALL CORE ON TAPE
277C 0000   107 STOP     HLT           HALT, TAPE IS DONE
277D 177C   108          JMP   STOP    IF RUN IS PUSHED
            109 *
            110 * MAG TAPE SECTION
            111 *
277E 0140   112 MAG      CXA
277F 97EB   113          LDX   CORE
2780 0501   114          DXS   1
```

```
2781 7800   115          STW * 0      SAVE RETURN
2782 0220   116 STAT1    DIN   2,0    MAG TAPE STATUS
2783 0810   117          SAP          CONTROLLER BUSY?
2784 1782   118          JMP   STAT1  YES, WAIT
2785 0A11   119          SLL   1
2786 0810   120          SAP          TAPE UNIT BUSY?
2787 1782   121          JMP   STAT1  YES, WAIT
2788 0A13   122          SLL   3
2789 0810   123          SAP          IS TAPE REWINDING?
278A 1782   124          JMP   STAT1  YES, WAIT
278B 87E8   125          LDW   MEM
278C 0321   126          DOT   2,1    SET MEMORY START ADDRESS
278D 87E7   127          LDW   WC     NUMBER OF WORDS
278E 0324   128          DOT   2,4    SET WORD COUNT, WRITE REC.
278F 0220   129 STAT2    DIN   2,0    MAG TAPE STATUS
2790 0A17   130          SLL   7
2791 0820   131          SAM          DONE WRITING A RECORD?
2792 178F   132          JMP   STAT2  NO, WAIT
2793 E7E4   133          AND   X0180  KEEP BITS 14,15
2794 0800   134          SAZ
2795 17B2   135          JMP   ERR    RATE OR TAPE ERROR
2796 97EB   136          LDX   CORE
2797 0501   137          DXS   1
2798 9800   138          LDX * 0      GET RETURN
2799 1800   139          JMP * 0      DO THE RETURN
279A 0080   140 RING     SMB   0
279B 206E   141          JSX   X'6E'
279C CEC5   142          TEXT  'NEED'
279D C5C4
279E 0080   143          SMB   0
279F 206E   144          JSX   X'6E'
27A0 D7D2   145          TEXT  'WRIT'
27A1 C9D4
27A2 0080   146          SMB   0
27A3 206E   147          JSX   X'6E'
27A4 D2C9   148          TEXT  'RING'
27A5 CEC7
27A6 0000   149          HLT           MOUNT TAPE, PUSH RUN
27A7 1748   150          JMP   STAT4
27A8 0080   151 CERR     SMB   0
27A9 206E   152          JSX   X'6E'
27AA C2C1   153          TEXT  'BAD '
27AB C4A0
27AC 0080   154          SMB   0
27AD 206E   155          JSX   X'6E'
27AE C3D3   156          TEXT  'CSUM'
27AF D5CD
27B0 0000   157          HLT           REDO MAG TAPE
27B1 17A8   158          JMP   CERR    IF RUN IS PUSHED
27B2 0080   159 ERR      SMB   0
27B3 206E   160          JSX   X'6E'
27B4 D4C1   161          TEXT  'TAPE'
27B5 D0C5
27B6 0080   162          SMB   0
27B7 206E   163          JSX   X'6E'
27B8 C5D2   164          TEXT  'EROR'
27B9 CFD2
```

```
27BA 0000   165          HLT            TAPE ERROR, REDO
27BB 17B2   166          JMP    ERR     IF RUN IS PUSHED
            167 *
            168 * RESTART SECTION
            169 *
27BC 0000   170 SMB      HLT            (LOADER HALTS HERE)
27BD 0089   171 RESTART  SMB    $
27BE 0050   172          SGM
27BF 27D5   173          JSX    CSUM    CALCULATE CHECK SUM
27C0 97EB   174          LDX    CORE
27C1 F800   175          CMW * 0
27C2 0860   176          SEQ            DO CHECK SUMS AGREE?
27C3 17A8   177          JMP    CERR    NO
27C4 97EB   178          LDX    CORE
27C5 87ED   179          LDW    SF
27C6 7800   180          STW * 0        RESTORE CORE END
27C7 0501   181          DXS    1
27C8 87EC   182          LDW    SE
27C9 7800   183          STW * 0        RESTORE CORE END -1
27CA 0080   184 RET      SMB    0
27CB 2083   185          JSX    X'83'   RETURN TO C. FORTRAN
            186 *
            187 * LOADER SECTION
            188 *
27CC 0100   189 LOADER   CLR
27CD 0089   190          SMB    SMB
27CE 77BC   191          STW    SMB
27CF 0321   192          DOT    2,1     START ADDRESS = 0
27D0 8008   193          LDW    8       NUMBER OF WORDS IN CORE
27D1 0326   194          DOT    2,6     READ RECORD
27D2 0089   195          SMB    SMB     GO TO HALT
27D3 17BC   196          JMP    SMB     JUST BEFORE RESTART
27D4 0000   197 CSIZE    D      0       # OF WORDS IN CORE
            198 *
            199 * CHECK SUM ROUTINE, ANSWER IN ACR.
            200 *
27D5 0140   201 CSUM     CXA
27D6 97EB   202          LDX    CORE
27D7 0501   203          DXS    1
27D8 7800   204          STW * 0        SAVE RETURN
27D9 0100   205          CLR
27DA 0501   206 AGAIN    DXS    1
27DB 17E0   207          JMP    ADDUP
27DC 97EB   208          LDX    CORE
27DD 0501   209          DXS    1
27DE 9800   210          LDX * 0        GET RETURN
27DF 1800   211          JMP * 0        DO THE RETURN
27E0 A800   212 ADDUP    ADD * 0        SUM CORE (EXCEPT LAST 2)
27E1 17DA   213          JMP    AGAIN
            214 *                       END OF CHECK SUM
27E2 0001   215 D1       D      1
27E3 0009   216 D9       D      9
27E4 0180   217 X0180    D      X'180'
27E5 1400   218 X1400    D      X'1400'
27E6 0D00   219 XD00     D      X'D00'
27E7 0000   220 WC       D      0
27E8 0000   221 MEM      D      0
```

```
27E9 27CC   222 LLOAD   D      LOADER
27EA 0000   223 NUMP    DATA   0
27EB 0000   224 CORE    D      0
27EC 0000   225 SE      D      0
27ED 0000   226 SF      D      0
            227         END
```

NO ERRORS

ADDUP	27E0	AGAIN	27DA	CERR	27A8	CONT	2743
CONT1	276F	CONT2	273A	CORE	27EB	CSIZE	27D4
CSUM	27D5	D1	27E2	D9	27E3	ERR	27B2
LLOAD	27E9	LOADER	27CC	MAG	277E	MEM	27E8
NUM	2731	NUMP	27EA	RESTART	27BD	RET	27CA
RING	279A	SE	27EC	SF	27ED	SKIP	2759
SMB	27BC	SS1	273D	SS2	2740	STAT1	2782
STAT2	278F	STAT3	2763	STAT4	2748	STEM	2730
STOP	277C	WC	27E7	X0180	27E4	X1400	27E5
XD00	27E6						

PAS?

```
                    1 ' 12/20/74:4 'DATA'
                    2         ORIG  X'2800'
                    3 * PROGRAM BRINGS IN DATA FROM INTERFACE
                    4 * AND SENDS DATA FROM A DESIGNATED
                    5 * CHANNEL TO THE DIGITAL DISPLAY ON THE
                    6 * INTERFACE.
                    7 *
                    8 * IN FORTRAN USE:
                    9 *       CALL DATA(JWIND,JRAW1,JRAW2,JCH)
                   10 * WHERE
                   11 *       JWIND = DATA FROM INTERFACE WIND CHANNEL
                   12 *       JRAW1 = DATA FROM INTERFACE CH1
                   13 *       JRAW2 = DATA FROM INTERFACE CH2
                   14 *       JCH   = 0 (SEND WIND TO DIGITAL DISPLAY)
                   15 *             = 1 (SEND CH1 DATA TO DIGITAL DISPLAY)
                   16 *             = 2 (SEND CH2 DATA TO DIGITAL DISPLAY)
                   17 *             > 2 (SEND NO DATA TO DIGITAL DISPLAY)
                   18 *
2800 0082          19 DATA    DATA  X'82'
2801 0000          20 JWIND   D     0
2802 0000          21 JRAW1   D     0
2803 0000          22 JRAW2   D     0
2804 0000          23 JCH     D     0
2805 03F6          24         DOT   15,6      MASTER RESET TO FIFO
2806 806C          25         LDW   DINT
2807 0080          26         SMB   0
2808 7015          27         STW   X'15'
2809 0080          28         SMB   0
280A 2083          29         JSX   X'83'     RETURN TO C. FORTRAN
                   30 *
                   31 * INTERRUPT LEVEL 5
                   32 *
280B 008A          33 INT     SMB   TX
280C 6069          34         STX   TX
280D 706A          35         STW   TA
280E 02F2          36         DIN   15,2      BRING IN DATA FROM FIFO
280F 7068          37         STW   T
2810 0A0E          38         SRL   14
2811 F065          39         CMW   D1
2812 0860          40         SEQ             IS DATA FROM WIND?
2813 101D          41         JMP   A         NO
2814 8068          42         LDW   T         YES
2815 E06B          43         AND   X3FF      EXTRACT WIND DATA
2816 9001          44         LDX   JWIND
2817 7800          45         STW * 0         STORE WIND DATA
2818 9004          46         LDX   JCH
2819 8800          47         LDW * 0
281A 0800          48         SAZ             DISPLAY WIND VALUE?
281B 1047          49         JMP   RET       NO
281C 1033          50         JMP   DISP      YES
281D F067          51 A       CMW   D2
281E 0860          52         SEQ             IS DATA FROM CH1?
281F 102A          53         JMP   B         NO
2820 8068          54         LDW   T         YES
2821 E06B          55         AND   X3FF      EXTRACT CH1 DATA
2822 9002          56         LDX   JRAW1
2823 7800          57         STW * 0         STORE CH1 DATA
```

```
2824 9004     58          LDX   JCH
2825 8800     59          LDW * 0
2826 F065     60          CMW   D1
2827 0860     61          SEQ               DISPLAY CH1 DATA?
2828 1047     62          JMP   RET         NO
2829 1033     63          JMP   DISP        YES
282A 8068     64 B        LDW   T
282B E06B     65          AND   X3FF        EXTRACT CH2 DATA
282C 9003     66          LDX   JRAW2
282D 7800     67          STW * 0           STORE CH2 DATA
282E 9004     68          LDX   JCH
282F 8800     69          LDW * 0
2830 F067     70          CMW   D2
2831 0860     71          SEQ               DISPLAY CH2 DATA?
2832 1047     72          JMP   RET         NO
2833 8068     73 DISP     LDW   T
2834 E06B     74          AND   X3FF        EXTRACT DATA
              75 *
              76 * CONVERT TO BCD
              77 *
2835 0810     78          SAP               NEGATIVE VALUE?
2836 0110     79          CMP               YES, ABSOLUTE VALUE
2837 F066     80          CMW   D9999
2838 0890     81          SLE               TOO BIG?
2839 8066     82          LDW   D9999       YES, LIMIT VALUE
283A 204A     83          JSX   RM10        REMAINDER SUBROUTINE
283B 0000     84 DIG4     D     0           LEAST SIG. DIGIT
283C 204A     85          JSX   RM10
283D 0000     86 DIG3     D     0
283E 204A     87          JSX   RM10
283F 0000     88 DIG2     D     0
2840 0A14     89          SLL   4           MOST SIG. DIGIT IN ACR
2841 C03F     90          ORI   DIG2
2842 0A14     91          SLL   4
2843 C03D     92          ORI   DIG3
2844 0A14     93          SLL   4
2845 C03B     94          ORI   DIG4        PACKED BCD IN ACR
2846 03F2     95 SEND     DOT   15,2        SEND TO DISPLAY
2847 9069     96 RET      LDX   TX          RESTORE INDEX
2848 806A     97          LDW   TA          RESTORE ACR
2849 0015     98          INR   5           INTERRUPT RETURN
284A 7063     99 RM10     STW   ASAV
284B 0800    100          SAZ               CHECK FOR ZERO NUMERATOR
284C 104F    101          JMP   $+3
284D 7800    102          STW * 0           STUFF ZERO REMAINDER
284E 1801    103          JMP * 1
284F 0A11    104          SLL   1
2850 A063    105          ADD   ASAV
2851 7064    106          STW   TSAV
2852 0A04    107          SRL   4
2853 A064    108          ADD   TSAV
2854 0A04    109          SRL   4
2855 A064    110          ADD   TSAV
2856 0A04    111          SRL   4
2857 A064    112          ADD   TSAV
2858 A065    113          ADD   D1
2859 0A05    114          SRL   5
```

```
285A 7064   115          STW    TSAV      QUOTIENT
285B 0A13   116          SLL    3
285C A064   117          ADD    TSAV
285D A064   118          ADD    TSAV
285E 0110   119          CMP               -10*QUOTIENT
285F A063   120          ADD    ASAV
2860 7800   121          STW *  0         REMAINDER
2861 8064   122          LDW    TSAV      QUOTIENT IN ACR
2862 1801   123          JMP *  1
2863 0000   124 ASAV     D      0
2864 0000   125 TSAV     D      0
2865 0001   126 D1       D      1
2866 270F   127 D9999    D      9999
2867 0002   128 D2       D      2         (NOT IN BASIC BCD)
2868 0000   129 T        DATA   0
2869 0000   130 TX       D      0
286A 0000   131 TA       D      0
286B 03FF   132 X3FF     D      X'3FF'
286C 280B   133 DINT     D      INT
            134          END
```

N O ERRORS

A	281D	ASAV	2863	B	282A	D1	2865
D2	2867	D9999	2866	DATA	2800	DIG2	283F
DIG3	283D	DIG4	283B	DINT	286C	DISP	2833
INT	280B	JCH	2804	JRAW1	2802	JRAW2	2803
JWIND	2801	RET	2847	RM10	284A	SEND	2846
T	2868	TA	286A	TSAV	2864	TX	2869
X3FF	286B						

PAS?

```
                1 ' 12/20/74:3 'LFD'
                2         ORIG  X'2870'
                3 * OUTPUTS LINE FEED TO TTY (OR TEK TERMINAL)
                4 * IN FORTRAN USE:
                5 *         CALL LFD
                6 *
2870 0082       7 LFD     DATA  X'82'
2871 02E0       8 A       DIN   14,0      BRING IN TEK STATUS
2872 0810       9         SAP             IS TEK STILL CONNECTED?
2873 1071      10         JMP   A
2874 03EA      11 B       DOT   14,10     SELECT TEK TERMINAL
2875 807E      12         LDW   LF        LOAD LINE FEED
2876 03EE      13         DOT   14,14     SEND DATA TO TEK TERMINAL
2877 02E0      14 C       DIN   14,0      TEK STATUS
2878 0A17      15         SLL   7
2879 0820      16         SAM             DONE YET?
287A 1077      17         JMP   C         NO
287B 03E0      18 D       DOT   14,0      DISCONNECT TEK TERMINAL
287C 0080      19         SMB   0
287D 2083      20         JSX   X'83'     GO BACK TO C. FORTRAN
287E 008A      21 LF      DATA  X'8A'
               22         END
```

NO ERRORS

```
A          2871  B          2874  C          2877  D          287B
LF         287E  LFD        2870
PAS?
```

```
                 1 ' 11/6/74:5 'POWF5'
                 2         ORIG  X'2890'
                 3 *
                 4 * PROGRAM IS A POWER FAIL SAFE ROUTINE.
                 5 * SET POWER FAIL SAFE IN FORTRAN PROGRAM BY:
                 6 *       CALL POWF5(NPF)
                 7 * WHERE NPF IS THE NUMBER OF POWER INTERRUPTS
                 8 * THAT OCCURED.
                 9 *
                10 * LOCATION X'4B' IS THE INTERRUPT ENABLE
                11 * STATUS WORD.
                12 * BITS 0 - 11 OF LOC X'4D' SET = #
                13 * OF POWER FAILURES.
                14 * BITS 12-15 OF
                15 * LOCATION X'4D' ARE SET TO 1,2,3,4 TO
                16 * SHOW HOW FAR THE POWERFAIL SAFE
                17 * ROUTINE GOT.
                18 * THIS PROGRAM CAN RESTART TTY & MAG TAPE IO
                19 * IF A POWER FAILURE OCCURS.
                20 *
     0007       21 INT      EQU   7
2890 0082       22 POWF     DATA  X'82'
2891 0000       23 NPF      D     0
2892 9091       24          LDX   NPF
2893 0100       25          CLR
2894 7101       26          STW   TPF      SET TPF = 0
2895 7800       27          STW * 0        SET NPF = 0
2896 8102       28          LDW   D1
2897 0080       29          SMB   0
2898 704D       30          STW   X'4D'    SET = 1
2899 8109       31          LDW   DSMBH
289A 0080       32          SMB   0
289B 701D       33          STW   INT+INT+INT+INT+1
289C 0027       34          ENB   INT
289D 0080       35          SMB   X'83'
289E 2083       36          JSX   X'83'  RETURN TO FORTRAN PROGRAM
                37 *
                38 * POWER DOWN SEQUENCE
                39 *
289F 008A       40 SMBH     SMB   $
28A0 00A0       41          MSK           MASK INTERRUPTS
28A1 60FE       42          STX   PFSX    SAVE INDEX VALUE
28A2 70FD       43          STW   PFSA    SAVE ACC. VALUE
                44 *
                45 *                      DISCONNECT ALL
                46 *                      PERIPHERAL DEVICES
                47 *
28A3 9106       48          LDX   X10     GET DEVICE COUNT
28A4 8107       49          LDW   X300    DOT 0,0
28A5 70A6       50          STW   $+1     SET TO EXECUTE
28A6 0000       51          D     0
28A7 A106       52          ADD   X10     GET TO NEXT DEVICE
28A8 0501       53          DXS   1       DEC COUNT
28A9 10A5       54          JMP   $-4     NOT DONE
28AA 8801       55          LDW * 1       GET CELL 0 (IX = FFFF)
28AB 70FF       56          STW   PFS0    SAVE VALUE IN LOC 0
28AC 809F       57          LDW   SMBH
```

```
28AD 0080     58          SMB   0
28AE 707D     59          STW   X'7D'    PUT SMB IN LOC X'7D'
28AF 80C1     60          LDW   JMPH
28B0 0080     61          SMB   0
28B1 707E     62          STW   X'7E'    PUT POWER UP JMP IN LOC X'7E'
28B2 8108     63          LDW   X107D
28B3 0080     64          SMB   0
28B4 7000     65          STW   0     STORE JMP TO X'7D' IN LOC 0
28B5 9091     66          LDX   NPF
28B6 8800     67          LDW * 0
28B7 A102     68          ADD   D1
28B8 7800     69          STW * 0       NPF=NPF+1
28B9 8101     70          LDW   TPF
28BA A102     71          ADD   D1
28BB 7101     72          STW   TPF     TPF = TPF + 1
28BC 0A14     73          SLL   4        MOVE 4 BITS TO LEFT
28BD A103     74          ADD   D2      SHOWS PFS GOT TO HERE
28BE 0080     75          SMB   0
28BF 704D     76          STW   X'4D'    RESTORE PF STATUS
28C0 0000     77          HLT           WAIT FOR A.C. POWER UP
28C1 10C2     78 JMPH     JMP   $+1
              79 *
              80 * POWER UP SEQUENCE
              81 *
28C2 0027     82          ENB   INT     ENABLE POWER FAIL SAFE
28C3 0080     83          SMB   0
28C4 804D     84          LDW   X'4D'     LOAD PF STATUS
28C5 E10A     85          AND   XFFF0   KEEP # OF POWER FAILS
28C6 A104     86          ADD   D3     SHOWS PFS GOT TO HERE
28C7 0080     87          SMB   0
28C8 704D     88          STW   X'4D'     SET = 3
28C9 0050     89          SGM
28CA 80FF     90          LDW   PFS0
28CB 0080     91          SMB   0
28CC 7000     92          STW   0       RESTORE LOCATION 0
28CD 03F6     93          DOT   15,6    MASTER RESET FIFO
28CE 0080     94          SMB   X'4B'
28CF 804B     95          LDW   X'4B'
28D0 E10B     96          AND   X400    EXTRACT INT. ENABLE 5 STATUS
28D1 0800     97          SAZ
28D2 0025     98          ENB   5
28D3 0080     99          SMB   0
28D4 905C    100          LDX   X'5C'   FIND END OF CORE
28D5 0507    101          DXS   7
28D6 8804    102          LDW * 4
28D7 710C    103          STW   IORET   SAVE IO RETURN
28D8 9800    104          LDX * 0
28D9 6100    105          STX   SAVE    SAVE LOC OF LAST FIOT
28DA 8800    106          LDW * 0       WORD 0 OF FIOT
28DB 0820    107          SAM           POWER DOWN DURING IO?
28DC 10F4    108          JMP   INRR    NO, NORMAL INR
28DD E10D    109          AND   X7FFF   YES
28DE 7800    110          STW * 0       SET BUSY BIT = 0
28DF 8802    111          LDW * 2       WORD 2 OF FIOT
28E0 7111    112          STW   WORD2   SAVE WORD 2 OF FIOT
28E1 E10E    113          AND   X1F0    EXTRACT UNIT NUMBER
28E2 F112    114          CMW   XD0
```

```
28E3 0860   115          SEQ          WAS DEVICE = TTY?
28E4 10F0   116          JMP   RETURN   NO
28E5 8111   117          LDW   WORD2
28E6 E10F   118          AND   X1FF    KEEP BITS 6-15
28E7 A110   119          ADD   X1800   PUT DISCONNECT IN JMP CODE
28E8 7802   120          STW * 2       CHANGE WORD 2 OF FIOT
28E9 0020   121          ENB   0       ALLOW TTY TO INTERRUPT
28EA 8113   122          LDW   CR
28EB 03EE   123          DOT   14,14   SEND CARRIAGE RETURN
28EC 02E0   124 WAIT     DIN   14,0    TTY STATUS
28ED 0A17   125          SLL   7
28EE 0820   126          SAM
28EF 10EC   127          JMP   WAIT
28F0 910C   128 RETURN   LDX   IORET   PRETEND IO WAS FINISHED
28F1 0501   129          DXS   1
28F2 0A10   130          NOP
28F3 1800   131          JMP * 0       GO BACK TO DOIO
28F4 0080   132 INRR     SMB   0
28F5 804D   133          LDW   X'4D'     LOAD PF STATUS
28F6 E10A   134          AND   XFFF0   KEEP # OF POWER FAILS
28F7 A105   135          ADD   D4      SHOWS IO WAS NOT IN PROGRESS
28F8 0080   136          SMB   0       WHEN THE POWER FAILED
28F9 704D   137          STW   X'4D'
28FA 90FE   138          LDX   PFSX    RESTORE INDEX
28FB 80FD   139          LDW   PFSA    RESTORE ACC.
28FC 0017   140          INR   INT
28FD 0000   141 PFSA     D     0
28FE 0000   142 PFSX     D     0
28FF 0000   143 PFS0     D     0
2900 0000   144 SAVE     D     0
2901 0000   145 TPF      D     0
2902 0001   146 D1       D     1
2903 0002   147 D2       D     2
2904 0003   148 D3       D     3
2905 0004   149 D4       D     4
2906 0010   150 X10      D     X'10'
2907 0300   151 X300     D     X'300'
2908 107D   152 X107D     D     X'107D'
2909 289F   153 DSMBH     D     SMBH
290A FFF0   154 XFFF0     D     X'FFF0'
290B 0400   155 X400      D     X'400'
290C 0000   156 IORET     D     0
290D 7FFF   157 X7FFF     D     X'7FFF'
290E 01F0   158 X1F0      D     X'1F0'
290F 01FF   159 X1FF      D     X'1FF'
2910 1800   160 X1800     D     X'1800'
2911 0000   161 WORD2     D     0
2912 00D0   162 XD0       D     X'D0'
2913 008D   163 CR        D     X'8D'
            164           END
```

NO ERRORS

CR	2913	D1	2902	D2	2903	D3	2904
D4	2905	DSMBH	2909	INRR	28F4	INT	0007
IORET	290C	JMPH	28C1	NPF	2891	PFS0	28FF
PFSA	28FD	PFSX	28FE	POWF	2890	RETURN	28F0
SAVE	2900	SMBH	289F	TPF	2901	WAIT	28EC
WORD2	2911	X10	2906	X107D	2908	X1800	2910
X1F0	290E	X1FF	290F	X300	2907	X400	290B
X7FFF	290D	XD0	2912	XFFF0	290A		

PAS?

```
                1 ' 10/31/74:5 'MULT1'
                2         ORIG  X'291B'
291B 0082       3 MULT1   D     X'82'
291C 0000       4 NVAL    D     0
291D 0000       5 JWIND   D     0
291E 0000       6 JCOUNT  D     0
291F 0000       7 JDONE   D     0
2920 0000       8 JEROR   D     0
2921 0000       9 JSKIP   D     0
               10 * 'MULT1' PROGRAM BRINGS IN WIND
               11 * DATA AND CH1 CH2 DATA FROM
               12 * MUX FIFO SYSTEM VIA INTERRUPT
               13 * LEVEL 5.
               14 * IN FORTRAN USE:
               15 *      CALL MULT1(K(1),JWIND,JCOUNT,JDONE,JEROR,JSKIP)
               16 * K(1) IS FIRST ELEMENT IN ARRAY
               17 * JWIND IS CURRENT WIND VALUE.
               18 * JCOUNT INDICATES THE ARRAY
               19 * ELEMENT CURRENTLY FILLED.
               20 * JDONE=1 WHEN ARRAY IS FULL.
               21 * JEROR = COUNT OF NUMBER OF BAD
               22 * DATA VALUES
               23 * JSKIP = NUMBER OF SAMPLES SKIPPED DUE
               24 * TO MISSING CHANNEL NUMBERS.
               25 * IN FORTRAN SET JDONE, JSKIP, AND
               26 * JCOUNT = 0 BEFORE STARTING
               27 * TO FILL ARRAY WITH DATA.
2922 008A      28         SMB   $
2923 0100      29         CLR
2924 9120      30         LDX   JEROR
2925 7800      31         STW * 0       SET JEROR = 0
2926 9121      32         LDX   JSKIP
2927 7800      33         STW * 0        SET JSKIP = 0
2928 818E      34         LDW   =2
2929 718D      35         STW   SW       SET SW = 2
292A 811C      36         LDW   NVAL
292B B18F      37         SUB   =1
292C 718C      38         STW   N      SET N=BEGINNING OF ARRAY - 1
292D 03F6      39         DOT   15,6  MASTER RESET TO MUX FIFO
292E 8190      40         LDW   =MULT5
292F 0080      41         SMB   X'15'
2930 7015      42         STW   X'15'
2931 0080      43         SMB   X'83'
2932 2083      44         JSX   X'83'
               45 * INTERRUPT LEVEL 5
2933 008A      46 MULT5   SMB   TX
2934 618A      47         STX   TX
2935 7189      48         STW   TA
2936 02F2      49         DIN   15,2    BRING IN DATA FROM MUX FIFO
2937 718B      50         STW   T
2938 0A0E      51         SRL   14
2939 F18F      52         CMW   =1
293A 0860      53         SEQ
293B 1141      54         JMP   A
293C 818B      55         LDW   T
293D E191      56         AND   =X'03FF'   EXTRACT WIND DATA
293E 911D      57         LDX   JWIND
```

```
293F 7800    58          STW * 0         STORE WIND DATA
2940 1163    59          JMP   RET
2941 F18E    60 A        CMW   =2
2942 0860    61          SEQ
2943 1166    62          JMP   B
2944 818D    63          LDW   SW
2945 F18E    64          CMW   =2
2946 0860    65          SEQ
2947 1149    66          JMP   REFIL2
2948 1153    67          JMP   NRM2
2949 9121    68 REFIL2   LDX   JSKIP     MISSED CHANNEL
294A 8800    69          LDW * 0
294B A18F    70          ADD   =1
294C 7800    71          STW * 0         JSKIP=JSKIP+1
294D 818C    72          LDW   N
294E 0130    73          CAX             POINTER IN INDEX
294F 818B    74          LDW   T
2950 E191    75          AND   =X'03FF'
2951 7800    76          STW * 0         OVERWRITE LAST CH DATA
2952 1163    77          JMP   RET
2953 818C    78 NRM2     LDW   N
2954 A18F    79          ADD   =1
2955 718C    80          STW   N         N=N+1
2956 0130    81          CAX             STORE PT. ADDRESS IN INDEX R.
2957 818B    82          LDW   T
2958 E191    83          AND   =X'03FF'  EXTRACT CH1 DATA
2959 7800    84          STW * 0         STORE CH1 DATA IN ARRAY
295A 8192    85          LDW   =3
295B 718D    86          STW   SW        SET SW = 3
295C 911E    87 UPCNT    LDX   JCOUNT
295D 8800    88          LDW * 0
295E A18F    89          ADD   =1
295F 7800    90          STW * 0         COUNT=COUNT+1
2960 F193    91          CMW   =1199
2961 0890    92          SLE
2962 117D    93          JMP   DONE      ARRAY IS FULL
2963 918A    94 RET      LDX   TX
2964 8189    95          LDW   TA
2965 0015    96          INR   5
2966 F192    97 B        CMW   =3
2967 0860    98          SEQ
2968 1184    99          JMP   ERROR
2969 818D   100          LDW   SW
296A F192   101          CMW   =3
296B 0860   102          SEQ
296C 116E   103          JMP   REFIL3
296D 1173   104          JMP   NRM3
296E 9121   105 REFIL3   LDX   JSKIP
296F 8800   106          LDW * 0
2970 A18F   107          ADD   =1
2971 7800   108          STW * 0         JSKIP=JSKIP+1
2972 1163   109          JMP   RET
2973 818C   110 NRM3     LDW   N
2974 A18F   111          ADD   =1
2975 718C   112          STW   N         N=N+1
2976 0130   113          CAX             STORE PT. ADDRESS IN INDEX R.
2977 818B   114          LDW   T
```

```
2978 E191   115          AND   =X'03FF'  EXTRACT CH2 DATA
2979 7800   116          STW * 0         STORE CH2 DATA IN ARRAY
297A 818E   117          LDW   =2
297B 718D   118          STW   SW        SET SW = 2
297C 115C   119          JMP   UPCNT
297D 811C   120 DONE     LDW   NVAL
297E B18F   121          SUB   =1
297F 718C   122          STW   N       SET N=BEGINNING OF ARRAY - 1
2980 911F   123          LDX   JDONE
2981 818F   124          LDW   =1
2982 7800   125          STW * 0       SET JDONE=1 (ARRAY IS FULL)
2983 1163   126          JMP   RET
2984 9120   127 ERROR    LDX   JEROR
2985 8800   128          LDW * 0
2986 A18F   129          ADD   =1     JEROR=JEROR+1
2987 7800   130          STW * 0      JEROR = COUNT OF BAD DATA
2988 1163   131          JMP   RET
2989 0000   132 TA       D     0
298A 0000   133 TX       D     0
298B 0000   134 T        D     0
298C 0000   135 N        D     0
298D 0000   136 SW       D     0
            137          END

298E 0002
298F 0001
2990 2933
2991 03FF
2992 0003
2993 04AF


NO ERRORS
```

A	2941	B	2966	DONE	297D	ERROR	2984
JCOUNT	291E	JDONE	291F	JEROR	2920	JSKIP	2921
JWIND	291D	MULTI	291B	MULT5	2933	N	298C
NRM2	2953	NRM3	2973	NVAL	291C	REFIL2	2949
REFIL3	296E	RET	2963	SW	298D	T	298B
TA	2989	TX	298A	UPCNT	295C		

PAS?

```
                1 ' 10/31/74:3 'MMON'
                2         ORIG  X'299B'
                3 * MMON MODIFIES THE MONITOR SO THAT
                4 * IT WILL NOT WRITE OUT M0 M0 ON THE TTY
                5 * CONTINUOUSLY WHEN THE TAPE UNIT GOES
                6 * DOWN.
                7 * IF UNIT GOES DOWN IT WILL TYPE:
                8 * TAPE
                9 * UNIT
               10 * DOWN
               11 * LOAD
               12 * PROG
               13 *
               14 * THEN IT WILL HALT WITH THE NUMBER
               15 * OF POWER FAILURES IN THE ACCUMULATOR
               16 * IN FORTRAN USE:
               17 *       CALL MMON(LOC)
               18 * WHERE LOC = DECIMAL VALUE OF MONITOR
               19 * LOCATION TO BE REPLACED WITH SMB & JMP.
               20 *
               21 * THE TWO LINES IN MONITOR TO BE REPLACED
               22 * ARE AS FOLLOWS:
               23 *   STW * 0      AND PLUG INTO MESSAGE
               24 *   JMP   TYPE   TYPE WILL RETURN TO CALLER
               25 *
299B 0082      26 MMON    DATA  X'82'
299C 0000      27 LOC     D     0
299D 919C      28         LDX   LOC
299E 9800      29         LDX * 0
299F 81A5      30         LDW   SMB
29A0 7800      31         STW * 0       STORE SMB IN MONITOR
29A1 81A6      32         LDW   JMP
29A2 7801      33         STW * 1       STORE JMP IN MONITOR
29A3 0080      34         SMB   0
29A4 2083      35         JSX   X'83'   RETURN TO C. FORTRAN
29A5 008A      36 SMB     SMB   A
29A6 11A7      37 JMP     JMP   A
29A7 0080      38 A       SMB   0
29A8 206E      39         JSX   X'6E'
29A9 D4C1      40         TEXT  'TAPE'
29AA D0C5
29AB 0080      41         SMB   0
29AC 206E      42         JSX   X'6E'
29AD D5CE      43         TEXT  'UNIT'
29AE C9D4
29AF 0080      44         SMB   0
29B0 206E      45         JSX   X'6E'
29B1 C4CF      46         TEXT  'DOWN'
29B2 D7CE
29B3 0080      47         SMB   0
29B4 206E      48         JSX   X'6E'
29B5 CCCF      49         TEXT  'LOAD'
29B6 C1C4
29B7 0080      50         SMB   0
29B8 206E      51         JSX   X'6E'
29B9 D0D2      52         TEXT  'PROG'
29BA CFC7
```

```
29BB 0080     53          SMB   0
29BC 804D     54          LDW   X'4D'
29BD 0A04     55          SRL   4
29BE 0000     56          HLT             RELOAD PROGRAM
              57          END
```

NO ERRORS

```
A       29A7  JMP       29A6  LOC       299C  MMON      299B
SMB     29A5
PAS?
```

```
                    1 ' 10/31/74:2 'DMAMAG'
                    2         ORIG  X'29C0'
                    3 * DMAMAG CHECKS DMA MAG TAPE STATUS
                    4 * IT SETS:
                    5 * MAG = 1 ON LINE, WRITE ENABLED, NOT ON BOT
                    6 * MAG = 2 ON LINE, NOT WRITE ENABLED, BOT
                    7 * MAG = 3 ON LINE, WRITE ENABLED, BOT
                    8 * MAG = 13 TAPE UNIT NOT READY
                    9 *           UNIT TURNED OFF OR
                   10 *           OFF LINE OR REWINDING
                   11 * MAG = 16 NOT WRITE ENABLED, EOT
                   12 * MAG = 17 WRITE ENABLED, EOT
                   13 * IN FORTRAN USE:
                   14 *       CALL DMAMAG(MAG)
29C0 0082          15 DMAMAG  DATA  X'82'
29C1 0000          16 MAG     D     0
29C2 008A          17          SMB   $
29C3 0220          18          DIN   2,0      GET MAG TAPE STATUS
29C4 0A02          19          SRL   2
29C5 E1D5          20          AND   =1
29C6 71D4          21          STW   T       SAVE BIT 13
29C7 0220          22          DIN   2,0      GET MAG TAPE STATUS
29C8 0A0C          23          SRL   12
29C9 E1D6          24          AND   =7       SAVE BITS 0,1&2
29CA C1D4          25          ORI   T        COMBINE BITS 0,1,2,13
29CB 71D4          26          STW   T
29CC 0220          27          DIN   2,0      GET MAG TAPE STATUS
29CD 0A08          28          SRL   8        MOVE BIT 3 OVER
29CE E1D7          29          AND   =X'10'
29CF C1D4          30          ORI   T        COMBINE BITS 3,0,1,2,13
29D0 91C1          31          LDX   MAG
29D1 7800          32          STW * 0
29D2 0080          33          SMB   X'83'
29D3 2083          34          JSX   X'83'
29D4 0000          35 T        D     0
                   36          END

29D5 0001
29D6 0007
29D7 0010
```

NO ERRORS

```
DMAMAG    29C0  MAG       29C1  T         29D4
PAS?
```

```
                1 ' 10/31/74:1  'INTL'
                2          ORIG   X'29E0'
29E0 0082       3 INT      D      X'82'
29E1 0000       4 LEV      DATA   0
29E2 0000       5 N        D      0
                6 * INT PROGRAM DISABLES OR ENABLES
                7 * INTERRUPT LEVEL LEV
                8 * FOR N = 1 DISABLE LEV
                9 * FOR N <> 1 ENABLE LEV
               10 * LOCATION X'4B' CONTAINS ENABLE
               11 * STATUS FOR ALL 15 INTERRUPTS
               12 * IE BIT=1 IF INTERRUPT IS ENABLED
               13 * THIS STATUS WORD IS USED IN
               14 * POWERFAIL SAFE PROGRAMS TO REENABLE
               15 * THE PROPER INTERRUPTS.
               16 * IN FORTRAN USE:
               17 *       CALL INTL(LEV,N)
               18 * WHERE LEV = 0 TO 15
29E3 0080      19          SMB    X'4B'
29E4 804B      20          LDW    X'4B'     LOAD INTERRUPT ENABLE STATUS
29E5 7210      21          STW    STAT
29E6 91E2      22          LDX    N
29E7 8800      23          LDW  * 0
29E8 F214      24          CMW    =1
29E9 0860      25          SEQ
29EA 11FF      26          JMP    A
29EB 91E1      27          LDX    LEV
29EC 8800      28          LDW  * 0
29ED F215      29          CMW    =15
29EE 0890      30          SLE
29EF 0000      31          HLT          BAD LEV VALUE
29F0 C211      32          ORI    F
29F1 71F2      33          STW    B     MODIFY DSB INSTRUCTION
29F2 0030      34 B        DSB    0     DISABLE LEV INTERRUPT
29F3 91E1      35          LDX    LEV
29F4 8800      36          LDW  * 0
29F5 C212      37          ORI    G
29F6 71F8      38          STW    C     MODIFY SRC INSTRUCTION
29F7 8216      39          LDW    =X'7FFF'
29F8 0A40      40 C        SRC    0           SHIFT 0 ALONG
29F9 E210      41          AND    STAT    SET BIT LEV = 0
29FA 0080      42 R        SMB    X'4B'
29FB 704B      43          STW    X'4B'     RESTORE INT. ENABLE STATUS
29FC 00B0      44          UNM               UNMASK POWERFAIL SAFE
29FD 0080      45          SMB    X'83'
29FE 2083      46          JSX    X'83'
29FF 00A0      47 A        MSK               MASK POWERFAIL SAFE
2A00 91E1      48          LDX    LEV
2A01 8800      49          LDW  * 0
2A02 F215      50          CMW    =15
2A03 0890      51          SLE
2A04 0000      52          HLT          BAD LEV VALUE
2A05 C213      53          ORI    H
2A06 7207      54          STW    D     MODIFY ENB INSTRUCTION
2A07 0020      55 D        ENB    0     ENABLE LEV INTERRUPT
2A08 91E1      56          LDX    LEV
2A09 8800      57          LDW  * 0
```

```
2A0A C212     58          ORI   G
2A0B 720D     59          STW   E     MODIFY SRC INSTRUCTION
2A0C 8217     60          LDW   =X'8000'
2A0D 0A40     61 E        SRC   0       SHIFT 1 ALONG
2A0E C210     62          ORI   STAT    SET BIT LEV = 1
2A0F 11FA     63          JMP   R
2A10 0000     64 STAT     D     0
2A11 0030     65 F        DSB   0
2A12 0A40     66 G        SRC   0
2A13 0020     67 H        ENB   0
              68          END

2A14 0001
2A15 000F
2A16 7FFF
2A17 8000
```

NO ERRORS

A	29FF	B	29F2	C	29F8	D	2A07
E	2A0D	F	2A11	G	2A12	H	2A13
INT	29E0	LEV	29E1	N	29E2	R	29FA
STAT	2A10						
PAS?							

```
                    1 ' 12/20/74:6 'DISP'
                    2 *       ORIG  X'2A20'
                    3 *
                    4 * WHEN CALLED BY FORTRAN IT CHECKS TO
                    5 * SEE IF VALUE OF JCHAN (CHANNEL #)
                    6 * AGREES WITH THE CHANNEL SWITCH SETTING
                    7 * (SSE) CHANGES K FROM BINARY TO BCD
                    8 * AND SENDS THE VALUE TO THE LED DISPLAY.
                    9 * IF JCHAN = 0 NO NEED TO CHECK SSE
                   10 * JUST CONVERT TO BCD AND SEND TO DISPLAY.
                   11 *
                   12 * IN FORTRAN USE:
                   13 *       CALL DISP(K,JCHAN)
                   14 * WHERE
                   15 * JCHAN = 0  CONVERT TO BCD, SEND TO DISPLAY
                   16 *       = 1  CHECK SSE, IF OK TO BCD & DISPLAY
                   17 *       = 2  CHECK SSE, IF OK TO BCD & DISPLAY
                   18 *
2A20 0082          19 DISP    DATA  X'82'
2A21 0000          20 K       D     0
2A22 0000          21 JCHAN   D     0
2A23 9222          22         LDX   JCHAN
2A24 8800          23         LDW * 0          GET CHANNEL NUMBER
2A25 0800          24         SAZ              NEED TO TEST CHANNEL #?
2A26 1228          25         JMP   TEST       YES
2A27 1231          26         JMP   BEGIN      NO, TO BCD TO DISPLAY
2A28 08B0          27 TEST    SSE
2A29 122E          28         JMP   TRUE       SWITCH SET TO CHAN 1
2A2A F264          29         CMW   D2         SWITCH SET TO CHAN 2
2A2B 0860          30         SEQ              DATA FROM CHAN 2?
2A2C 1245          31         JMP   RET        NO
2A2D 1231          32         JMP   BEGIN      YES, CONVERT TO BCD
2A2E F262          33 TRUE    CMW   D1
2A2F 0860          34         SEQ              DATA FROM CHAN 1?
2A30 1245          35         JMP   RET        NO
2A31 9221          36 BEGIN   LDX   K
2A32 8800          37         LDW * 0          LOAD VALUE TO BE DISPLAYED
                   38 *
                   39 * CONVERT TO BCD
                   40 *
2A33 0810          41         SAP              NEGATIVE VALUE?
2A34 0110          42         CMP              YES, ABSOLUTE VALUE
2A35 F263          43         CMW   D9999
2A36 0890          44         SLE              TOO BIG?
2A37 8263          45         LDW   D9999      YES, LIMIT VALUE
2A38 2247          46         JSX   RM10       REMAINDER SUBROUTINE
2A39 0000          47 DIG4    D     0          LEAST SIG. DIGIT
2A3A 2247          48         JSX   RM10
2A3B 0000          49 DIG3    D     0
2A3C 2247          50         JSX   RM10
2A3D 0000          51 DIG2    D     0
2A3E 0A14          52         SLL   4          MOST SIG. DIGIT IN ACR
2A3F C23D          53         ORI   DIG2
2A40 0A14          54         SLL   4
2A41 C23B          55         ORI   DIG3
2A42 0A14          56         SLL   4
2A43 C239          57         ORI   DIG4       PACKED BCD IN ACR
```

```
2A44 03F2    58 SEND     DOT    15,2      SEND TO DISPLAY
2A45 0080    59 RET      SMB    0
2A46 2083    60          JSX    X'83'     RETURN TO C. FORTRAN
2A47 7260    61 RM10     STW    ASAV
2A48 0800    62          SAZ              CHECK FOR ZERO NUMERATOR
2A49 124C    63          JMP    $+3
2A4A 7800    64          STW *  0         STUFF ZERO REMAINDER
2A4B 1801    65          JMP *  1
2A4C 0A11    66          SLL    1
2A4D A260    67          ADD    ASAV
2A4E 7261    68          STW    TSAV
2A4F 0A04    69          SRL    4
2A50 A261    70          ADD    TSAV
2A51 0A04    71          SRL    4
2A52 A261    72          ADD    TSAV
2A53 0A04    73          SRL    4
2A54 A261    74          ADD    TSAV
2A55 A262    75          ADD    D1
2A56 0A05    76          SRL    5
2A57 7261    77          STW    TSAV      QUOTIENT
2A58 0A13    78          SLL    3
2A59 A261    79          ADD    TSAV
2A5A A261    80          ADD    TSAV
2A5B 0110    81          CMP              -10*QUOTIENT
2A5C A260    82          ADD    ASAV
2A5D 7800    83          STW *  0         REMAINDER
2A5E 8261    84          LDW    TSAV      QUOTIENT IN ACR
2A5F 1801    85          JMP *  1
2A60 0000    86 ASAV     D      0
2A61 0000    87 TSAV     D      0
2A62 0001    88 D1       D      1
2A63 270F    89 D9999    D      9999
2A64 0002    90 D2       D      2         (NOT IN BASIC BCD)
             91          END
```

NO ERRORS

```
A SAV    2A60  BEGIN    2A31  D1       2A62  D2       2A64
D 9999   2A63  DIG2     2A3D  DIG3     2A3B  DIG4     2A39
D ISP    2A20  JCHAN    2A22  K        2A21  RET      2A45
R M10    2A47  SEND     2A44  TEST     2A28  TRUE     2A2E
T SAV    2A61
P AS?
```

```
      12/11/74:8 'LEADR'                                        PAGE    1

                 1 * 12/11/74:8 'LEADR'
                 2         ORIG  X'2A20'
                 3 * PUNCHES PAPER TAPE LEADER WHEN CALLED
                 4 * BY MAIN PROGRAM.
                 5 *
                 6 * IN FORTRAN USE:
                 7 *         CALL LEADR
                 8 *
  2A20 0082      9 LEADR   DATA  X'82'
       0005     10 UNIT    EQU   5
  2A21 0080     11         OPEN  F,BUF,WC,UNIT,X'E',1
S 2A22 2042
  2A23 2A37
  2A24 2A40
  2A25 2A3F
  2A26 0005
  2A27 000E
  2A28 8001
  2A29 823D     12         LDW   F+6
  2A2A C236     13         ORI   X8000      SET SPECIAL FORMAT
  2A2B 723D     14         STW   F+6
  2A2C 0080     15         DOIO  F,BUF,WC
S 2A2D 2044
  2A2E 2A37
  2A2F 2A40
  2A30 AA3F
  2A31 0080     16         STAT  F
S 2A32 2046
  2A33 AA37
  2A34 0080     17         SMB   0
  2A35 2083     18         JSX   X'83'      RETURN TO C. FORTRAN
  2A36 8000     19 X8000   D     X'8000'
  2A37          20 F       RES   8
  2A3F 0023     21 WC      DATA  35
  2A40          22 BUF     RES   35
       0042     23 OPEN    EQU   66
       0044     24 DOIO    EQU   68
       0046     25 STAT    EQU   70
                26         END


NO ERRORS
```

```
BUF     2A40  DOIO     0044  F      2A37  LEADR  2A20
OPEN    0042  STAT     0046  UNIT   0005  WC     2A3F
X8000   2A36
PAS?
```

```
                    1 * 11/6/74:8 'CLOK'
                    2         ORIG  X'2A70'
                    3 * PROGRAM READS NEW CMOS CLOCK UNTIL
                    4 * TIME AGREES IN TWO SUCCESSIVE READS.
                    5 *
                    6 * IN FORTRAN USE:
                    7 *       CALL CLOK(JH,JM,JS)
                    8 * WHERE:
                    9 *       JH = HOURS
                   10 *       JM = MINUTES
                   11 *       JS = SECONDS
                   12 *
2A70 0082           13 CLOK    DATA  X'82'
2A71 0000           14 JH      D     0
2A72 0000           15 JM      D     0
2A73 0000           16 JS      D     0
2A74 02F3           17         DIN   15,3    BRING IN M1, S10, S1
2A75 72AB           18         STW   TM1
2A76 02F4           19         DIN   15,4    BRING IN H10, H1, M10
2A77 72AD           20         STW   TH1
2A78 02F3           21 NEW     DIN   15,3
2A79 72AC           22         STW   TM2
2A7A 02F4           23         DIN   15,4
2A7B 72AE           24         STW   TH2
2A7C F2AD           25         CMW   TH1
2A7D 0860           26         SEQ             SAME TIME?
2A7E 1283           27         JMP   MOVE    TIMES DO NOT AGREE
2A7F 82AC           28         LDW   TM2
2A80 F2AB           29         CMW   TM1
2A81 0870           30         SNE             SAME TIME?
2A82 1288           31         JMP   DONE    TIMES AGREE
2A83 82AC           32 MOVE    LDW   TM2
2A84 72AB           33         STW   TM1
2A85 82AE           34         LDW   TH2
2A86 72AD           35         STW   TH1
2A87 1278           36         JMP   NEW     READ TIME AGAIN
2A88 82AC           37 DONE    LDW   TM2
2A89 229D           38         JSX   COMB    COMBINE SECONDS DIGITS
2A8A 9273           39         LDX   JS
2A8B 7800           40         STW * 0       STORE TOTAL SECONDS
2A8C 82AE           41         LDW   TH2
2A8D 0A04           42         SRL   4
2A8E 229D           43         JSX   COMB    COMBINE HOURS DIGITS
2A8F 9271           44         LDX   JH
2A90 7800           45         STW * 0       STORE TOTAL HOURS
2A91 82AC           46         LDW   TM2
2A92 0A08           47         SRL   8
2A93 72B1           48         STW   T1
2A94 82AE           49         LDW   TH2
2A95 E2AF           50         AND   XF
2A96 0A14           51         SLL   4
2A97 A2B1           52         ADD   T1
2A98 229D           53         JSX   COMB    COMBINE MINUTES DIGITS
2A99 9272           54         LDX   JM
2A9A 7800           55         STW * 0       STORE TOTAL MINUTES
2A9B 0080           56         SMB   0
2A9C 2083           57         JSX   X'83'   RETURN TO C. FORTRAN
```

```
2A9D 62B0     58 COMB     STX   HERE   BCD TO DECIMAL VALUE
2A9E 72B1     59          STW   T1
2A9F E2AF     60          AND   XF
2AA0 72B2     61          STW   T2     SAVE UNITS VALUE
2AA1 82B1     62          LDW   T1
2AA2 0A04     63          SRL   4
2AA3 E2AF     64          AND   XF
2AA4 72B1     65          STW   T1
2AA5 0A13     66          SLL   3
2AA6 A2B1     67          ADD   T1
2AA7 A2B1     68          ADD   T1     X10  (TENS VALUE)
2AA8 A2B2     69          ADD   T2     TOTAL VALUE
2AA9 92B0     70          LDX   HERE
2AAA 1800     71          JMP * 0      BACK TO MAIN ROUTINE
2AAB 0000     72 TM1      D     0
2AAC 0000     73 TM2      D     0
2AAD 0000     74 TH1      D     0
2AAE 0000     75 TH2      D     0
2AAF 000F     76 XF       D     X'F'
2AB0 0000     77 HERE     D     0
2AB1 0000     78 T1       D     0
2AB2 0000     79 T2       D     0
              80          END
```

NO ERRORS

```
CLOK    2A70  COMB    2A9D  DONE    2A88  HERE    2AB0
JH      2A71  JM      2A72  JS      2A73  MOVE    2A83
NEW     2A78  T1      2AB1  T2      2AB2  TH1     2AAD
TH2     2AAE  TM1     2AAB  TM2     2AAC  XF      2AAF
PAS?
```

7. APPENDIX C. PROGRAM LISTINGS FOR INTERFACE TEST PROGRAMS AND REAL-TIME ANALYZER ASSEMBLY LANGUAGE DRIVER

Appendix C contains program listings for "MINOS1", TCLOK", TRTA1", and "RTA1".

```
                1 ' 11/26/74:4 'MINOS1'
                2         ORIG  X'1800'
                3 * MINIMAL TEST OF COMMUNITY NOISE SYSTEM.
                4 * CHANGE LOC B TO GIVE DIFFERENT DELAY
                5 * TIME BETWEEN INTERRUPT AND THE DIN IN.
                6 * USE SS1 AND SS2 TO GIVE THE FOLLOWING
                7 * VALUES ON EXTERNAL DISPLAY:
                8 *
                9 * WIND    SS1, SS2 DOWN    (ID = A)
               10 * CH1     SS1 UP           (ID = B)
               11 * CH2     SS2 UP           (ID = C)
               12 *
               13 * EXTERNAL SENSE UP FOR DECIMAL DISPLAY
               14 * EXTERNAL SENSE DOWN FOR HEX DISPLAY
               15 *    (IN HEX MODE MOST SIGNIFICANT DIGIT
               16 *     IS ID)
               17 *
               18 * PUT SS0 UP TO HALT AFTER DISPLAY
               19 * PUT SS3 UP TO GIVE MASTER RESET
               20 *                AFTER DISPLAY
               21 *
               22 *
               23 * IF THE WORD 'FACE' REMAINS ON THE DISPLAY
               24 * IT SAYS THAT:
               25 *                1) DISPLAY IS WORKING
               26 *                2) LEVEL 5 DATA INTERRUPT
               27 *                   FROM INTERFACE IS
               28 *                   MISSING OR CPU INTERRUPT
               29 *                   CARD IS MALFUNCTIONING
               30 *
1800 0086      31 START   SMB   $
1801 0050      32         SGM
1802 806E      33         LDW   =IT
1803 0080      34         SMB   0
1804 7015      35         STW   X'15'
1805 03F6      36         DOT   15,6      MASTER RESET TO FIFO
1806 803C      37         LDW   FACE
1807 03F2      38         DOT   15,2     SEND FACE TO DISPLAY
1808 0025      39         ENB   5
1809 1009      40 A       JMP   A
180A 9035      41 IT      LDX   B
180B 0501      42 C       DXS   1          DELAY
180C 100B      43         JMP   C
180D 02F2      44         DIN   15,2       BRING IN FIFO DATA
180E 7037      45         STW   T
180F 08D0      46         SS1
1810 1029      47         JMP   S1
1811 08E0      48         SS2
1812 102F      49         JMP   S2
1813 0A0E      50         SRL   14
1814 806F      51         CMW   =1
1815 0860      52         SEQ           WIND DATA?
1816 1028      53         JMP   RETA      NO
1817 8037      54 DISP    LDW   T
1818 E03B      55         AND   X3FF
1819 08B0      56         SSE          WANT HEX OR DECIMAL?
181A 1022      57         JMP   JBCD      DECIMAL
```

```
181B 7038    58          STW   T2      SAVE VALUE
181C 8037    59          LDW   T
181D E039    60          AND   XC000
181E 0A02    61          SRL   2
181F A03A    62          ADD   X9000   A, B, OR C ID
1820 A038    63          ADD   T2      TOTAL HEX NUMBER
1821 1023    64          JMP   SEND
1822 203D    65 JBCD     JSX   BCD     CONVERT TO BCD
1823 03F2    66 SEND     DOT   15,2    SEND TO DISPLAY
1824 08C0    67          SS0
1825 0000    68          HLT           PUSH RUN TO CONTINUE
1826 08F0    69          SS3
1827 03F6    70          DOT   15,6    MASTER RESET TO FIFO
1828 0015    71 RETA     INR   5
1829 8037    72 S1       LDW   T
182A 0A0E    73          SRL   14
182B F070    74          CMW   =2
182C 0860    75          SEQ           DATA CHANNEL 1?
182D 1028    76          JMP   RETA    NO
182E 1017    77          JMP   DISP    YES
182F 8037    78 S2       LDW   T
1830 0A0E    79          SRL   14
1831 F071    80          CMW   =3
1832 0860    81          SEQ           DATA CHANNEL 2?
1833 1028    82          JMP   RETA    NO
1834 1017    83          JMP   DISP    YES
1835 0000    84 B        D     0
1836 0000    85 RSAVE    D     0
1837 0000    86 T        D     0
1838 0000    87 T2       D      0
1839 C000    88 XC000    D     X'C000'
183A 9000    89 X9000    D     X'9000'
183B 03FF    90 X3FF     D     X'3FF'
183C FACE    91 FACE     D     X'FACE'
             92 *
             93 * CONVERT TO BCD
             94 *
183D 6036    95 BCD      STX   RSAVE   SAVE RETURN
183E 0810    96          SAP           NEGATIVE VALUE?
183F 0110    97          CMP           YES, ABSOLUTE VALUE
1840 F06D    98          CMW   D9999
1841 0890    99          SLE           TOO BIG?
1842 806D   100          LDW   D9999   YES, LIMIT VALUE
1843 2051   101          JSX   RM10    REMAINDER SUBROUTINE
1844 0000   102 DIG4     D     0       LEAST SIG. DIGIT
1845 2051   103          JSX   RM10
1846 0000   104 DIG3     D     0
1847 2051   105          JSX   RM10
1848 0000   106 DIG2     D     0
1849 0A14   107          SLL   4       MOST SIG. DIGIT IN ACR
184A C048   108          ORI   DIG2
184B 0A14   109          SLL   4
184C C046   110          ORI   DIG3
184D 0A14   111          SLL   4
184E C044   112          ORI   DIG4    PACKED BCD IN ACR
184F 9036   113 RET      LDX   RSAVE   GET RETURN
1850 1800   114          JMP * 0
```

```
1851 706A   115 RM10    STW    ASAV
1852 0800   116         SAZ                 CHECK FOR ZERO NUMERATOR
1853 1056   117         JMP    $+3
1854 7800   118         STW  * 0           STUFF ZERO REMAINDER
1855 1801   119         JMP  * 1
1856 0A11   120         SLL    1
1857 A06A   121         ADD    ASAV
1858 706B   122         STW    TSAV
1859 0A04   123         SRL    4
185A A06B   124         ADD    TSAV
185B 0A04   125         SRL    4
185C A06B   126         ADD    TSAV
185D 0A04   127         SRL    4
185E A06B   128         ADD    TSAV
185F A06C   129         ADD    D1
1860 0A05   130         SRL    5
1861 706B   131         STW    TSAV         QUOTIENT
1862 0A13   132         SLL    3
1863 A06B   133         ADD    TSAV
1864 A06B   134         ADD    TSAV
1865 0110   135         CMP                 -10*QUOTIENT
1866 A06A   136         ADD    ASAV
1867 7800   137         STW  * 0           REMAINDER
1868 806B   138         LDW    TSAV         QUOTIENT IN ACR
1869 1801   139         JMP  * 1
186A 0000   140 ASAV    D      0
186B 0000   141 TSAV    D      0
186C 0001   142 D1      D      1
186D 270F   143 D9999   D      9999
     1800   144         END    START

186E 180A
186F 0001
1870 0002
1871 0003
```

NO ERRORS

A	1809	ASAV	186A	B	1835	BCD	183D
C	180B	D1	186C	D9999	186D	DIG2	1848
DIG3	1846	DIG4	1844	DISP	1817	FACE	183C
IT	180A	JBCD	1822	RET	184F	RETA	1828
RM10	1851	RSAVE	1836	S1	1829	S2	182F
SEND	1823	START	1800	T	1837	T2	1838
TSAV	186B	X3FF	183B	X9000	183A	XC000	1839

PAS?

```
 1 C 10/30/74:4 'TCLOK'
 2 C       TEST CMOS 'CLOK' PROGRAM
 3 C      PUT SS0 UP TO STOP PRINTOUT.
 4        LOCATE CLOK,X'2A70'
 5        WRITE(3,10)
 6 10     FORMAT(3/,'PUT SS0 UP TO STOP PRINTOUT')
 7         WRITE (3,1)
 8 1       FORMAT (5/,' HR MIN SEC',/)
 9 3       CALL CLOK (JH,JM,JS)
10         WRITE (3,2) JH,JM,JS
11 2       FORMAT (I3,I4,I4)
12 4       KK = ISWCH(0) + 1
13         GO TO (3,4),KK
           END
```

```
C 1/27/75:6 'TRTA1'
C TESTS RTA1 PROGRAM & PRINTS OUT ALL 1/3 OCTAVE
C VALUES, WEIGHTED VALUE, LINEAR VALUE.
C
      LOCATE RTA1,X'2400'
      DIMENSION KVAL(40),KUND(40),B(33)
      B(1)=12.5
      B(2)=16.
      B(3)=20.
      B(4)=25.
      B(5)=31.5
      B(6)=40.
      B(7)=50.
      B(8)=63.
      B(9)=80.
      B(10)=100.
      B(11)=125.
      B(12)=160.
      B(13)=200.
      B(14)=250.
      B(15)=315.
      B(16)=400.
      B(17)=500.
      B(18)=630.
      B(19)=800.
      B(20)=1000.
      B(21)=1250.
      B(22)=1600.
      B(23)=2000.
      B(24)=2500.
      B(25)=3150.
      B(26)=4000.
      B(27)=5000.
      B(28)=6300.
      B(29)=8000.
      B(30)=10000.
      B(31)=12500.
      B(32)=16000.
      B(33)=20000.
      WRITE(3,1)
1     FORMAT(2/,'PUT SS0 UP TO BRING IN DATA FROM RTA.',3/)
2     KK=ISWCH(0)+1
      GO TO(2,3),KK
3     KVAL(40)=0
```

```
45          CALL RTA1(KVAL(1),KUND(1),JOVER)
46          WRITE(3,4)
47  4       FORMAT(4/,' REAL - TIME 1/3 OCTAVE ANALYZER TYPE 3347 OUTPUT',
48          WRITE(3,5)
49  5       FORMAT(3X,'BAND',5X,'DB',3X,'UFLOW',5X,'BAND',5X,'DB',3X,'UFL
50          AVAL=FLOAT(KVAL(1))/10.
51          BVAL=FLOAT(KVAL(17))/10.
52          WRITE(3,6)AVAL,KUND(1),B(16),BVAL,KUND(17)
53  6       FORMAT(/,' BASE L.',F7.1,I5,4X,F8.1,F7.1,I5)
54          DO 10 J=2,16
55          JJ=J+16
56          AVAL=FLOAT(KVAL(J))/10.
57          BVAL=FLOAT(KVAL(JJ))/10.
58  10      WRITE(3,11)B(J-1),AVAL,KUND(J),B(JJ-1),BVAL,KUND(JJ)
59  11      FORMAT(F8.1,F7.1,I5,4X,F8.1,F7.1,I5)
60          AVAL=FLOAT(KVAL(38))/10.
61          BVAL=FLOAT(KVAL(33))/10.
62          WRITE(3,12)AVAL,KUND(38),B(32),BVAL,KUND(33)
63  12      FORMAT(' WEIGHT.',F7.1,I5,4X,F8.1,F7.1,I5)
64          AVAL=FLOAT(KVAL(39))/10.
65          BVAL=FLOAT(KVAL(34))/10.
66          WRITE(3,13)AVAL,KUND(39),B(33),BVAL,KUND(34)
67  13      FORMAT(' LINEAR ',F7.1,I5,4X,F8.1,F7.1,I5)
68          IF(JOVER)23,22,23
69  23      WRITE(3,25)
70  25      FORMAT(/,'RTA INPUT AMPLIFIER WAS OVERLOADED.')
71  22      IF(KVAL(40))20,2,20
72  20      WRITE(3,21)KVAL(40)
73  21      FORMAT(/,'EXTRA VALUE FROM RTA.  KVAL(40) = ',I5)
74          GO TO 2
            END
```

```
                1 * 1/27/75:2 'RTA1'
                2         ORIG  X'2400'
                3 *
                4 * WHEN CALLED BY C. FORTRAN MAIN PROGRAM IT BRINGS
                5 * IN 39 VALUES FROM THE REAL TIME ANALYZER.
                6 *
                7 * IN C. FORTRAN USE:
                8 *       DIMENSION KVAL(39),KUND(39)
                9 *       CALL RTA(KVAL(1),KUND(1),JOVER)
               10 * WHERE
               11 *       KVAL(1) = BASE LINE VALUE (DB X 10)
               12 *       KVAL(2) = 12.5 HZ VALUE (DB X 10)
               13 *         .
               14 *         .
               15 *         .
               16 *       KVAL(34) = 20000 HZ VALUE (DB X 10)
               17 *       KVAL(38) = WEIGHTED VALUE (DB X 10)
               18 *       KVAL(39) = LINEAR VALUE (DB X 10)
               19 *
               20 *       KUND IS 39 WORD ARRAY IDICATING IF
               21 *            A PARTICULAR 1/3 OCTAVE VALUE
               22 *            IS ABOVE THE BASE LINE OR EQUAL
               23 *            TO OR BELOW THE BASE LINE
               24 *            (CALLED UNDERANGE)
               25 * I.E.  KUND(11) = 0 (100 HZ BAND > BASE LINE)
               26 *       KUND(11) = 1 (100 HZ BAND = OR < BASE LINE)
               27 *
               28 *       JOVER = 0  NORMAL SIGNAL LEVEL
               29 *             = 1  INPUT AMPLIFIER IS OVERLOADED
               30 *                  (CHANGE INPUT ATTENUATOR SETTING
               31 *                  TO REDUCE SIGNAL LEVEL)
               32 *
2400 0082      33 RTA1    DATA  X'82'
2401 0000      34 KVAL    D     0
2402 0000      35 KUND    D     0
2403 0000      36 JOVER   D     0
2404 8432      37         LDW   DEND
2405 0080      38         SMB   0
2406 7009      39         STW   9          STUFF INTERRUPT 2 LINKAGE
2407 8433      40         LDW   DDATA
2408 0080      41         SMB   0
2409 700D      42         STW   X'D'       STUFF INTERRUPT 3 LINKAGE
240A 8401      43         LDW   KVAL
240B 7428      44         STW   COUNT1     LOCATION OF KVAL(1)
240C 8402      45         LDW   KUND
240D 7420      46         STW   COUNT2     LOCATION OF KUND(1)
240E 0100      47         CLR
240F 7434      48         STW   TJOVER     SET TJOVER = 0
2410 0023      49         ENB   3          ENABLE DATA READY INTERRUPT
2411 0361      50         DOT   6,1        DATA REQUEST TO RTA
2412 0022      51         ENB   2          ENABLE END OF SCAN INTERRUPT
2413 1413      52 WAIT    JMP   WAIT       WAIT FOR DATA READY INTERRUPT
               53 *
               54 * DATA READY INTERRUPT LEVEL 3
               55 *
2414 0261      56 DATARDY DIN   6,1        BRING IN DATA FROM RTA
2415 7435      57         STW   T
```

```
2416 0820     58          SAM             INPUT AMPLIFIER OVERLOADED?
2417 141A     59          JMP   CONT      NO
2418 8436     60          LDW   ONE       YES
2419 7434     61          STW   TJOVER    SET TJOVER = 1
241A 8435     62 CONT     LDW   T
241B 9420     63          LDX   COUNT2    IX POINTS TO KUND ELEMENT
241C E437     64          AND   X4000     EXTRACT UNDERANGE BIT
241D 0A0E     65          SRL   14        MOVE UNDERRANGE BIT
241E 7800     66          STW * 0         STORE IN KUND ARRAY
241F 0401     67          IXS   1
2420 0000     68 COUNT2   D     0
2421 6420     69          STX   COUNT2    COUNT2 = COUNT2 + 1
2422 8435     70          LDW   T
2423 E438     71          AND   X3FFF     EXTRACT BCD DATA
2424 2439     72          JSX   BCDB      CONVERT BCD TO BINARY
2425 9428     73          LDX   COUNT1
2426 7800     74          STW * 0         STORE BINARY IN KVAL ARRAY
2427 0401     75          IXS   1
2428 0000     76 COUNT1   D     0
2429 6428     77          STX   COUNT1    COUNT1 = COUNT1 + 1
242A 0013     78          INR   3         DATA INTERRUPT RETURN
              79 *
              80 * END OF SCAN INTERRUPT LEVEL 2
              81 *
242B 0033     82 ENDSCAN  DSB   3         DISABLE INTERRUPT 3
242C 0032     83          DSB   2         DISABLE INTERRUPT 2
242D 8434     84          LDW   TJOVER
242E 9403     85          LDX   JOVER
242F 7800     86          STW * 0         PASS JOVER VALUE TO FORTRAN
2430 0080     87          SMB   0
2431 2083     88          JSX   X'83'     RETURN TO C. FORTRAN
2432 242B     89 DEND     D     ENDSCAN
2433 2414     90 DDATA    D     DATARDY
2434 0000     91 TJOVER   D     0
2435 0000     92 T        D     0
2436 0001     93 ONE      D     1
2437 4000     94 X4000    D     X'4000'
2438 3FFF     95 X3FFF    D     X'3FFF'
              96 *
              97 * CONVERT BCD TO BINARY ROUTINE
              98 *
2439 6451     99 BCDB     STX   RT        SAVE RETURN
243A 7452    100          STW   BCDN
243B 0100    101          CLR
243C 7453    102          STW   TOT       ZERO BINARY NUMBER
243D 2446    103          JSX   MB10      DIG1 X 10
243E 2446    104          JSX   MB10      DIG2 X 10 + DIG1 X 100
243F 2446    105          JSX   MB10      DIG3X10+DIG2X100+DIG1X1000
2440 8452    106          LDW   BCDN
2441 0A54    107          SLC   4
2442 E454    108          AND   XF
2443 A453    109          ADD   TOT       +DIG4
2444 9451    110          LDX   RT
2445 1800    111          JMP * 0
2446 8452    112 MB10     LDW   BCDN
2447 0A54    113          SLC   4
2448 7452    114          STW   BCDN
```

```
2449 E454   115          AND   XF      NEW DIGIT
244A A453   116          ADD   TOT
244B 7453   117          STW   TOT
244C 0A13   118          SLL   3       X8
244D A453   119          ADD   TOT
244E A453   120          ADD   TOT     X10
244F 7453   121          STW   TOT     NEW TOTAL
2450 1800   122          JMP * 0
2451 0000   123 RT       D     0       RETURN LOC
2452 0000   124 BCDN     D     0       BCD STRING
2453 0000   125 TOT      D     0       BINARY TOTAL
2454 000F   126 XF       D     X'F'    MASK
            127          END
```

NO ERRORS

```
BCDB      2439  BCDN      2452  CONT      241A  COUNT1    2428
COUNT2    2420  DATARDY   2414  DDATA     2433  DEND      2432
ENDSCAN   242B  JOVER     2403  KUND      2402  KVAL      2401
MB10      2446  ONE       2436  RT        2451  RTA1      2400
T         2435  TJOVER    2434  TOT       2453  WAIT      2413
X3FFF     2438  X4000     2437  XF        2454
PAS?
```

8. APPENDIX D. FORTRAN SOFTWARE LISTINGS FOR THE ANALYSIS OF COMMUNITY NOISE DATA

Appendix D contains program listings for Tape Read Program and Calcomp Plot Program.

Tape Read Program

```
      C  PROGRAM TO READ 24 HOURS OF DATA AND PLOT L1 ETC.,LEQ EACH HOUR.
         DIMENSION IDAY(5),LINE(2,6)
         DIMENSION SPLF(24,6,2),SPECS(35)
         DIMENSION YF(100),XF(100),FIT(5)
         INTEGER DAYST,HRST
         DIMENSION NBIN(100,2),IC2(2),J3(2,10000)
         COMMON J1(2000),J2(10000)
         EQUIVALENCE(J1(1),J3(1,1))
         DATA(FIT(J),J=1,5)/1.,10.,50.,90.,99./
      C READ IN BEGINNING,END,DELTA FOR BIN - E.G. 40,80,2 MEANS BIN IN 20 BINS FROM
      C    .  40 TO 80 DB WITH 2 DB BIN WIDTH.
         READ(5,1) IBEG,IEND,IDEL
      C FORMAT() MEANS FREE FORMAT, ALL NUMBERS SEPARATED BY COMMAS.  E.G 40,80,2
        1 FORMAT()
         NUM=(IEND-IBEG)/IDEL
         WRITE(6,2) NUM,IBEG,IEND,IDEL
        2 FORMAT(1H),'FOLLOWING DATA IS FOR',I3,'BINS FROM',I3,' TO',I3,
       .'DB IN',I3,' DB STEPS')
      C READ IN STARTING DAY,HR,NUM OF HRS
         READ(5,9)DAYST,HRST,NHR
        9 FORMAT(3I6)
         READ(0,10)IDAY(2),IDAY(4)
       10 FORMAT(2A6)
         IDAY(1)='   DAY'
         IDAY(3)='  HOUR'
         IDAY(5)=-0
       50 CALL RDUNPK(7,L2)
         IF(J1(1201).EQ.DAYST.AND.J1(1202).EQ.HRST)GO TO 11
         GO TO 50
       11 DO 100 II=1,NHR
         CALL NTRAN(7,7,-1)
         DO 15 J=1,2
         IC2(J)=0
         DO 15 I=1,100
       15 NBIN(I,J)=0
       17 IHR=J1(1202)
       13 CALL RDUNPK(7,L2)
      C  L2 NEGATIVE MEANS READ ERROR. IGNORE BLOCK AND CONTINUE
         IF(L2.LT.0) GO TO 13
         IF(J1(1207).NE.0)GO TO 13
         IF(J1(1202).NE.IHR)GO TO 16
      C  TO 30 BINS DATA IN GIVEN RANGE. IGNORE DATA OUTSIDE RANGE.
      C  DATA GREATER THAN OR EQUAL TO 40 AND LESS THAN 41 PUT IN 40DB BIN.
         DO 30 J=1,2
         DO 30 I=1,600
         IF(J3(J,I)/10..LT.IBEG.OR.J3(J,I)/10..GT.IEND) GO TO 30
         IC2(J)=IC2(J)+1
         SPLF(II,6,J)=SPLF(II,6,J)+10**(J3(J,II/100.I
         DO 20 K=1,NUM
         TEST =IBEG+K*IDEL
       20 IF(J3(J,I)/10..LT. TEST) GO TO 21
       21 NBIN(K,J)=NBIN(K,J)+1
       30 CONTINUE
         GO TO 17
       16 DO 40 J=1,2
         IF(IC2(J).EQ.0) GO TO 40
         SPLF(II,6,J)=10*ALOG10(SPLF(II,6,J)/IC2(J)I
         XF(1)=NBIN(NUM,J)
         DO 31 K=2,NUM
         J4=NUM-K+1
       31 XF(K)=XF(K-1)+NBIN(J4,JI
         DO 41 K=1,NUM
         XF(K)=XF(K)/IC2(J)*100.
       41 YF(K)=IEND+1-K
      C  INTERPOLATE TO CALCULATE L1,L10 ETC.
         DO 53 L=1,5
       53 SPLF(1),L,J)=AITINT(XF,YF,NUM,FIT(L),2,TI)
       40 CONTINUE
```

```
  100 CONTINUE
C  SPECS ARRAY IS FOR GRAPHICAL DISPLAY SYSTEM(CALCOMP)
      SPECS(1)=1
      SPECS(2)=1
      SPECS(3)=NHR
      SPECS(4)=1
      SPECS(5)=IEND
      SPECS(6)=IBEG
      SPECS(7)=NHR-1
      SPECS(8)=8
      SPECS(11)=1
      SPECS(12)=8
      SPECS(13)=NHR
      SPECS(14)=1
      SPECS(15)=1
      SPECS(16)=1
      SPECS(17)=.1
      SPECS(18)=.1
      SPECS(19)=0
      SPECS(20)=0
      SPECS(21)=1
      LINE(1,1)=' L1'
      LINE(1,2)=' L10'
      LINE(1,3)=' L50'
      LINE(1,4)=' L90'
      LINE(1,5)=' L99'
      LINE(1,6)=' LEQ'
      DO 14 I=1,6
   14 LINE(2,I)=-0
      DO 12 I=1,NHR
   12 XF(I)=I
      J=1
      SPECS(24)=0
      SPECS(26)=0
      SPECS(9)=1
      SPECS(10)=1
      CALL GDLILI(SPECS)
      SPECS(9)=NHR-1
      SPECS(10)=4
      CALL NODLIB(SPECS)
      CALL NODLIL(SPECS)
      CALL TITLEB('HOURS AFTER START',SPECS)
      SPECS(22)=1.5
      SPECS(23)=8.5
      CALL TITLEG(1.,IDAY,SPECS)
      SPECS(22)=.2
      SPECS(23)=4.83
      DO 102 L=1,6
      SPECS(23)=SPECS(23)-.15
      SPECS(16)=L
      CALL SYMKEY(1.,LINE(1,L),SPECS)
      CALL PSLILI(XF,SPLF(1,L,J),SPECS)
      CALL SLLILI(XF,SPLF(1,L,J),SPECS)
  102 CONTINUE
      CALL GDSEND(SPECS)
      STOP
      END
```

```
      SUBROUTINE RDUNPK(M,L2)
      COMMON J1(2000),J2(1000)
      CALL NTRAN(M,2,600,J2,L2,22)
      IF(L2.GT.0)GO TO 15
      WRITE(6,1)L2
    1 FORMAT(1H ,2015)
      GO TO 99
   15 N=9*L2/4
      DO 10 I=1,N,9
      K=1+4*(I-1)/9
      J1(I)=FLD(0,16,J2(K))
      J1(I+1)=FLD(16,16,J2(K))
      J1(I+2)=FLD(32,4,J2(K))*16*16*16+FLD(0,12,J2(K+1))
      J1(I+3)=FLD(12,16,J2(K+1))
      J1(I+4)=FLD(28,8,J2(K+1))*16*16+FLD(0,8,J2(K+2))
      J1(I+5)=FLD(8,16,J2(K+2))
      J1(I+6)=FLD(24,12,J2(K+2))*16+FLD(0,4,J2(K+3))
      J1(I+7)=FLD(4,16,J2(K+3))
      J1(I+8)=FLD(20,16,J2(K+3))
   10 CONTINUE
   99 RETURN
      END
```

```
      C  PROGRAM TO READ DATA,BIN,CALCULATE L1,10,50,90,99,LEQ AND LIST.
         DIMENSION YF(100),XF(100),FIT(5),SPLF(5),XLEG(2)
         INTEGER DAYST,HRST,MINST,DAYFIN,HRFIN,MINFIN
         DIMENSION NBIN(100,2),IC2(2),J3(2,6001
         COMMON J1(1215),J2(1000)
         EQUIVALENCE(J1(1),J3(1,1))
         DATA(FIT(J),J=1,5)/1.,10.,50.,90.,99./
      C READ IN BEGINNING,END,DELTA FOR BIN - E.G. 40,80,2 MEANS BIN IN 20 BINS FROM
      C    .  40 TO 80 DB WITH 2 DB BIN WIDTH.
         READ(5,1) IBEG,IEND,IDEL
      C FORMAT() MEANS FREE FORMAT, ALL NUMBERS SEPARATED BY COMMAS.  E.G 40,80,2
        1 FORMAT()
         NUM=(IEND-IBEG)/IDEL
         WRITE(6,2) NUM,IBEG,IEND,IDEL
        2 FORMAT(1H1,'FOLLOWING DATA IS FOR',I3,'BINS FROM',I3,' TO',I3,
       .'DB IN',I3,' DB STEPS')
      C READ IN STARTING DAY,HR,MIN AND FINISHING DAY,HR,MIN.
         READ(5,1) DAYST,HRST,MINST,DAYFIN,HRFIN,MINFIN
         READ(5,1)ICTRL
       50 CALL RDUNPK(7,L2)
         IF(J1(12D1).EQ.DAYST.AND.J1(1202).EQ.HRST.AND.J1(1203).GE.MINSTIGO
       .TO 11
         GO TO 50
       11 CALL NTRAN(7,7,-11
       13 CALL RDUNPK(7,L2)
      C  L2 NEGATIVE MEANS READ ERROR. IGNORE BLOCK AND CONTINUE
         IF(L2.LT.0) GO TO 13
         IC1=IC1+1
         IF(ICTRL.EQ.1)WRITE(6,10)(J1(J),J=1201,1215)
         IF(ICTRL.EQ.2)WRITE(6,10)J1
       10 FORMAT(1H 2D15)
      C  TO 30 BINS DATA IN GIVEN RANGE. IGNORE DATA OUTSIDE RANGE.
      C  DATA GREATER THAN OR EQUAL TO 40 AND LESS THAN 41 PUT IN 40DB BIN.
         DO 30 J=1,2
         DO 30 I=1,600
         J3(J,I)=J3(J,II*10
         IF(J3(J,I)/10..LT.IBEG.OR.J3(J,II/10..GE.IENDI GO TO 30
         IC2(J)=IC2(J)+1
         XLEG(J)=XLEG(J)+10**(J3(J,II/100.)
         DO 20 K=1,NUM
         TEST =IBEG+K*IDEL
       20 IF(J3(J,I)/10..LT. TESTI GO TO 21
       21 NBIN(K,J)=NBIN(K,J)+1
       30 CONTINUE
         IF(J1(12D1).EQ.DAYFIN.AND.J1(1202).EQ.HRFIN.AND.J1(1203).GE.MINFIN
       .) GO TO 12
         GO TO 13
       12 IC1=600*IC1
         WRITE(6,3) DAYST,HRST,MINST
        3 FORMAT(1H ,'DATA STARTS ON DAY',I4,' HOUR',I3,' MINUTE',I3)
         WRITE(6,4) DAYFIN,HRFIN,MINFIN
        4 FORMAT(1H ,'DATA ENDS ON DAY',I4,' HOUR',I3,' MINUTE',I3)
         WRITE(6,5)IC1
        5 FORMAT(1H ,'TOTAL POINTS READ =',I10)
         DO 40 J=1,2
         WRITE(6,6)J,IC2(J)
        6 FORMAT(1HO,'FOR CHANNEL',I2,' TOTAL POINTS BINNED IN RANGE =',I10)
         IF(IC2(J).EQ.0) GO TO 40
         WRITE(6,7)
        7 FORMAT(1H ,' XLEQ    L1    L10   L50   L90   L99')
         XLEG(J)=10*ALOG10(XLEG(J)/IC2(J))
         XF(1)=NBIN(NUM,J)
         DO 31 K=2,NUM
         J4=NUM-K+1
       31 XF(K)=XF(K-1)+NBIN(J4,J)
         DO 41 K=1,NUM
         XF(K)=XF(K)/IC2(J)*100.
       41 YF(K)=IENO+1-K
      C  INTERPOLATE TO CALCULATE L1,L10 ETC.
         DO 53 L=1,5
       53 SPLF(L)=AITINT(XF,YF,NUM,FIT(L),2,T1)
         WRITE(6,8) XLEG(J),SPLF
        8 FORMAT(1H ,10F6.1)
       40 CONTINUE
         STOP
         END
```

ACKNOWLEDGEMENTS

The authors wish to express their appreciation to Dr. Robert Joseph at the U. S. Army Environmental Hygiene Agency, Aberdeen Proving Ground, Maryland for the loan of the Bio-Acoustic Division's minicomputer system during interface development and for his helpful suggestions at the time of software development.

We wish to thank Mr. James Heinen, Miss Toni Savoy and Mr. Mitchell Tarica for the many hours they spent constructing the interface hardware.

We want to express our gratitude to Mrs. Marilyn Cadoff for her aid in compiling this report.

We also thank Mrs. Janet Russell and Mrs. Margaret Hildebrand for lending their typing skills to the preparation of this document.

Finally, we wish to thank Mr. William Haight for his judicious review and commentary on this report.

NBS-114A (REV. 7-73)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET	1. PUBLICATION OR REPORT NO. NBSIR 75-692	2. Gov't Accession No.	3. Recipient's Accession No.

4. TITLE AND SUBTITLE

A Minicomputer-Based System for the Measurement and Analysis of Community Noise

5. Publication Date: April 1975

6. Performing Organization Code

7. AUTHOR(S)

R. L. Fisher, D. S. Blomquist, J. S. Forrer, and D. M. Corley

8. Performing Organ. Report No. NBSIR 75-692

9. PERFORMING ORGANIZATION NAME AND ADDRESS

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20234

10. Project/Task/Work Unit No. 2130454

11. Contract/Grant No.

12. Sponsoring Organization Name and Complete Address *(Street, City, State, ZIP)*

Bioacoustics Division
U.S. Army Environmental Hygiene Agency
Aberdeen Proving Ground, Maryland 21010

13. Type of Report & Period Covered Final 5/74-2/75

14. Sponsoring Agency Code

15. SUPPLEMENTARY NOTES

16. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)*

An operating system for the measurement and analysis of community noise was turned over to the Army for their use in February 1975, thus accomplishing the transfer of technology developed by NBS to the Bioacoustics Division, U. S. Army Environmental Hygiene Agency. This report documents the hardware and software packages prepared by NBS in support of this system.

17. KEY WORDS *(six to twelve entries; alphabetical order; capitalize only the first letter of the first key word unless a proper name; separated by semicolons)*

Community noise; computer interface; instrumentation; minicomputer; noise.

18. AVAILABILITY

[X] Unlimited

[ ] For Official Distribution. Do Not Release to NTIS

[ ] Order From Sup. of Doc., U.S. Government Printing Office Washington, D.C. 20402, SD Cat. No. C13

[X] Order From National Technical Information Service (NTIS) Springfield, Virginia 22151

19. SECURITY CLASS (THIS REPORT)	21. NO. OF PAGES
UNCLASSIFIED	105
20. SECURITY CLASS (THIS PAGE)	**22. Price**
UNCLASSIFIED	$5.50

USCOMM-DC 29042-P74